MASTER EN SCIENCES INFORMATIQUES

Des Systèmes Intégrés de Gestion de Réseaux. (SIGR)

Boulanger, Benoit

Award date:
1995

Awarding institution:
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique

Année académique 1994-1995

Des Systèmes Intégrés de Gestion de Réseaux. (SIGR)

Benoit Boulanger

Mémoire présenté en vue de l'obtention du grade de Licencié et Maître en Informatique

Résumé

Les réseaux et ressources informatiques sont devenus des éléments essentiels de la compétitivité des entreprises modernes; leur gestion doit faire l'objet d'une attention particulière. Ce mémoire porte sur un système utilisable pour faciliter et optimiser la surveillance et le contrôle des moyens de télécommunications: les **Systèmes Intégrés de Gestion de Réseaux** (SIGR). Il se compose de deux parties comprenant chacune deux chapitres.

La première partie se veut théorique. Le premier chapitre présente les principaux protocoles de gestion de réseau(x) en détail mais aussi simplement que possible. Le second définit le concept de SIGR et introduit différents aspects de la gestion intégrée de réseaux.

La seconde partie est quelque peu plus pratique. Le premier de ses chapitres décrit neuf produits présentés par leur constructeur comme étant des moyens de gestion intégrée. Le dernier chapitre de ce travail propose une méthode complète d'implantation d'un SIGR.

Abstract

Nowadays, networks and IT resources are essential elements for the competitiveness of modern enterprises; so their management is also important. This work concerns a system wich can be used to facilitate, an to optimize the control and the monitoring of telecommunications means. It's the **Integrated Network Management Systems** (INMS). This work is composed of two parts organised in two chapters.

The first part is a theoretical one. The first chapter presents the main management protocols with details but as simple as possible. The second one defines the INMS concept and introduce different aspects of integrated network management.

The second part is more practical, its first chapter describes nine products wich are presented by the constructor as a means usable for integrated management. The last chapter of this work propose a complete implementatin method of an INMS.

Préface.

Nous tenons à remercier monsieur le professeur van Bastelaer pour les nombreux et précieux conseils qu'il nous a prodigués tout au long de ce mémoire ainsi que madame Hoge-Nachtergaele, son assistante, pour sa collaboration.

Nous remercions monsieur Trinon de la société BIM pour les facilités offertes.

Parmi les constructeurs avec qui nous avons eu contact, nous désirons remercier tout particulièrement le personnel de chez ALCATEL pour son accueil chaleureux et ses précieuses informations. Bien entendu nous devons également remercier le personnel des sociétés BULL, SUN et DIGITAL qui a répondu nos sollicitations. Enfin, nous remercions encore le Capitaine de gendarmerie Eggermont pour sa précieuse relecture.

Table des matières

INTRODUCTION

Introduction.

Dans l'histoire informatique d'une entreprise de taille relativement importante, c.-à-d. comptant plusieurs sièges répartis dans un ou plusieurs pays, on peut vraisemblablement observer ce qui suit.

L'apparition des micro- et mini-ordinateurs a permis la banalisation de l'informatique. Petit à petit, chaque département a disposé de ses propres ressources informatiques (Etape 1 Figure I.1). La multiplication des mêmes programmes et fichiers de données représentait un réel gaspillage. On a donc regroupé les moyens d'un site au sein d'un LAN (Local Area Network) (Etape 2 Figure I.1). A partir de ce moment, il n'était plus suffisant que chaque utilisateur gère son poste de travail, mais il fallait prévoir une coordination des moyens individuels. Il était de plus nécessaire d'organiser la gestion des moyens permettant la communication entre les postes de travail du site. Les systèmes de gestion de réseau(x) locaux et d'éléments de réseau ont alors fait leur apparition. Ainsi, de véritables "îles d'automatisation" sont apparues. A côté de cela, les progrès technologiques réalisés dans d'autres domaines ont eu pour conséquence de voir se développer bon nombre d'appareils pilotés par des ordinateurs (centraux téléphoniques, système de climatisation, ...) (Etape 3 Figure I.1).

La communication au sein d'une entreprise devenant un avantage concurrentiel, le raisonnement fait lors de la création des LAN a été appliqué à l'ensemble de l'entreprise. Tous les LAN ont donc été rassemblés au sein d'un réseau à l'échelle de l'entreprise (Backbone ou WAN - Wide Area Network). Toutes les "îles" ont pu communiquer grâce à l'avènement des *bridges*, des *gateways* (passerelles), ... Un système de gestion des moyens du WAN a fait son apparition, mais chacun restait responsable de son LAN et de ses autres moyens automatisés (Etape 4 Figure I.1). Partant de la constatation que tous accomplissaient les mêmes tâches de gestion, on les a alors rassemblées au niveau d'un centre de contrôle des réseaux (et moyens automatisés). Les îles ont donc été fédérées.

Malheureusement, chacune des ressources gérées nécessite encore son propre système de gestion. L'opérateur du centre de contrôle se voit astreint à surveiller une multitude d'écrans différents correspondant chacun à un système de gestion spécifique (Etape 5 Figure I.1); sa tâche est ardue et la qualité de service reste relativement moyenne. L'idée qui vient alors est de permettre la gestion de l'ensemble des facilités à partir d'un seul système de gestion (Etape 6 Figure I.1). Celui-ci devra être capable de masquer la diversité des équipements et devra présenter les moyens gérés comme un tout cohérent. Ce système de gestion est appelé : **Système Intégré de Gestion de Réseaux** (SIGR ou INMS - Integrated Network Management System).

Objectif.

Comme nous venons de le décrire, il est clair que ce type de système présente un intérêt majeur pour toute entreprise décentralisée de taille importante. Dans ce travail, nous allons tenter d'en faire le tour. Nous ne rentrerons pas dans les détails mais tenterons d'en voir toutes les facettes. Notre objectif est de donner une base à tout informaticien confronté au problème d'implantation d'un SIGR.

Méthode.

Pour réaliser ce travail, nous avons tout d'abord dû améliorer nos connaissances en matière de protocoles de gestion. Nous avons ensuite pu aborder le problème de l'intégration, tout d'abord sous l'angle des télécommunications et ensuite du point de vue de son environnement. Nous avons encore étudié quelques produits en nous basant sur la documentation commerciale les concernant. Cela nous a permis de faire des observations que nous avons complétées de lectures pour en arriver à pouvoir proposer une méthode d'implantation.

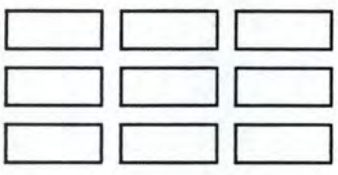
Divers.

Les points majeurs de ce travail sont certainement les chapitres 2 et 4.

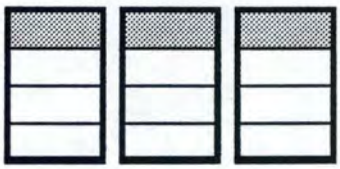
Dans le chapitre 2, nous situons le concept de gestion intégrée par rapport à la gestion de réseau(x). Nous y présentons le concept d'hétérogénéité qui est une des causes ayant entraîné le développement de systèmes intégrés. Nous définissons ensuite le concept de SIGR, tout d'abord au moyen d'une architecture générique et ensuite au travers de quatre modèles. Enfin, pour ouvrir des pistes de recherches dans des domaines connexes aux télécommunications mais liés aux SIGR, nous en parcourons l'environnement.

Dans le chapitre 4, nous proposons une méthode complète d'implantation d'un SIGR. Celle-ci est illustrée au moyen d'un cas fictif. Une attention particulière est accordée aux critères d'évaluation qui seront précisés dans un éventuel cahier des charges.

Pour aider le lecteur à découvrir ce travail et à en retirer l'information recherchée le plus aisément possible, nous proposons un plan de lecture en annexe A.



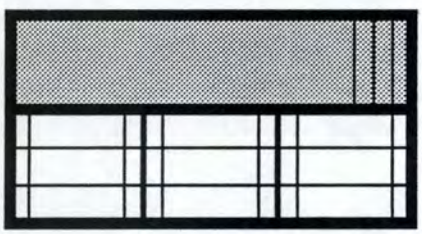
Etape 1



Etape 2

Etape 3

Etape 4



Etape 5

Etape 6

Légende:

= ressource automatisée

= micro- ou mini-ordinateur

= système de gestion

= LAN

= WAN ou Backbone

= Intégrateur

***Figure I.1** : Etapes menant à l'installation d'un SIGR.*

Partie 1: Protocoles et concepts.

Chapitre 1 : La gestion de réseau(x).

1.0. Introduction.

1.0.1. Gérer.

Avant de parler de la gestion de réseau(x) il nous paraît bon de rappeler ce que signifie gérer. Gérer sous-entend SURVEILLER, observer, et CONTROLER, diriger. Dans la gestion de réseau(x) on retrouvera ces deux activités. Il s'agit en fait de types d'activités.

La surveillance d'un réseau, c'est la surveillance de toutes ses opérations. C'est l'interrogation de tous les appareils gérables formant le réseau afin d'obtenir les valeurs de divers paramètres qui seront observées et analysées dans le but de découvrir tout dysfonctionnement. Cette observation doit également permettre une étude des performances.

Le contrôle d'un réseau, c'est en fait le contrôle des équipements critiques et de leur mode d'interaction. Il s'agit en fait d'agir sur la configuration des appareils gérables en modifiant les valeurs de certains paramètres; le but est d'avoir un réseau qui fonctionne d'une manière optimale par rapport à ses capacités et à nos objectifs. En général, on aura au préalable fait des analyses et simulations afin de connaître les valeurs à assigner aux paramètres.

Ainsi [TERP92] définit la gestion de réseau(x) comme *le déploiement et la coordination de ressources pour planifier, faire fonctionner, analyser, concevoir et étendre le réseau de communication afin de toujours remplir les objectifs en matière de niveau de service à un coût raisonnable et avec une capacité optimale.* Nous verrons plus tard que le monde des télécommunications subdivise l'activité de gestion en cinq domaines fonctionnels comprenant chacun des activités de surveillance et de contrôle.

1.0.2. Historique.

Comparée à l'introduction d'autres technologies, l'évolution de l'informatique au sein des entreprises a été fulgurante. Ainsi, après l'ère des mainframes et des centres de calcul, on a vu arriver les P.C. qui sont venus d'abord les compléter, mais qui petit à petit les ont reléguer aux traitements de masse. Cette prise de pouvoir des micro- et mini-ordinateurs n'a été rendue possible que par l'entremise des RESEAUX et des systèmes distribués. Aujourd'hui, les entreprises sont occupées à fédérer des îles d'automatisation créées hier avec les réseaux locaux.

L'importance du marché des réseaux a attiré de très nombreux constructeurs.Afin de permettre l'interconnexion des solutions qu'ils ont développées, il a bien fallu prévoir des standards et des protocoles. Deux grandes familles ont vu le jour: l'une, supportée par l'International Organization for Standardization (ISO) s'appuyant entre autres sur les travaux d'IBM proposait le modèle Open System Interconnection (OSI) et l'autre, sur l'initiative du Departement of Defense (DoD) et de son réseau ARPANET, proposait la suite de protocoles TCP/IP. Mais quelle était la situation au point de vue de la gestion de ces éléments? Etant donné que les réseaux étaient devenus un élément vital et une ressource critique pour l'entreprise, il fallait sérieusement penser à prendre en considération leur gestion [ALCA94] .

Fin des années 70, il n'existait pratiquement pas d'outils de gestion de réseau(x) dans le monde TCP/IP. On se contentait alors de l'outil Internet Control Message Protocol (ICMP); celui-ci permettait essentiellement de s'assurer de la liaison entre deux entités et de déterminer le délai de transmission du réseau. Il était à la base du programme Packet Internet Grouper (PING) qui, avec quelques outils supplémentaires, allait permettre de gérer les réseaux pendant quelques années. Fin des années 80, le nombre d'hôtes Internet croissant d'une manière exponentielle, ces outils n'ont plus suffi. Diverses pistes ont été suivies et l'Internet Activity Board (IAB*) a décidé, afin de répondre rapidement aux besoins, que le protocole Simple Network-Management Protocol (SNMP) serait développé comme solution à court terme et que le protocole Common Management Information Protocol (CMIP) over TCP/IP, soit CMOT, assurerait la compatibilité entre la gestion dans la suite TCP/IP et dans la suite OSI; il serait ainsi la solution à plus long terme. Notons encore que SNMP est en fait une évolution de Simple Gateway Management Protocol (SGMP). A l'origine SNMP et CMOT devaient fonctionner sur base de la même Structure d'Information de Gestion (Structure of Management Information - SMI). Cela apparut rapidement impossible. Il fut alors décidé que le développement de SNMP et de CMOT

* IAB = Comité international chargé de la supervision des développements des activités dans le domaine de la suite de protocoles Internet.

s'effectuerait en parallèle [STAL94]. Il est à noter qu'actuellement CMOT est très discret tandis que SNMP est largement implanté.

Dès leur proposition de standardisation au début des années 90, l'ISO et le CCITT (actuellement l'UIT-T ou Union Internationale des Télécommunications - Secteur de la normalisation des télécommunications) se sont penchés sur la gestion des réseaux. C'est d'ailleurs dans ce domaine qu'ils ont été les plus prolixes. Ils ont tout d'abord déterminé la structure de gestion pour le modèle OSI et ensuite, sur cette base, ils ont approfondi la question en spécifiant les divers éléments de la structure. Le document de base (ISO 7498-4 ou X.700) précise que le système de gestion doit permettre aux gestionnaires de :

- ☑ planifier, organiser, surveiller, contrôler et attribuer les services offerts par le réseau,
- ☑ répondre aux besoins des utilisateurs,
- ☑ permettre de prévoir le comportement du réseau,
- ☑ garantir la sécurité des informations.

Notons que l'UIT-T a réservé la série des recommandations X.700 à ce domaine.

Fin des années 80, IBM faisait avec ONA (Open Network- management Architecture) une proposition d'architecture pour la gestion de réseau(x). Celle-ci reposait sur leur réseau SNA mais devait également permettre la gestion de réseau(x) non SNA. En diffusant le format de leurs interfaces, ils tentaient d'imposer leur solution, comme ils l'avaient fait avec l'architecture SNA. [TERP92]

Nous terminerons en disant que le succès du système de gestion suivant OSI reste toujours un point d'interrogation. Les constructeurs ne peuvent certes plus l'ignorer, mais il s'agit maintenant de savoir si les utilisateurs feront le pas.

1.1. La gestion selon TCP/IP.

1.1.1. Concepts importants.

Dans la gestion suivant les principes développés par l'Internet Activities Board (IAB), TROIS concepts, reposant sur la notion *d'objet*, sont particulièrement importants. Ce sont :

① la Structure d'Information de Gestion (Structure of Management Information - SMI),

② la Base de Données des Informations de Gestion (Management Information Base-MIB),

③ le Protocole Simple de Gestion de Réseau(x) (Simple Network Management Protocol - SNMP).

Les maîtres mots des diverses recommandations en ce domaine sont ***simplicité et extensibilité***. Enfin, notons encore que toutes les structures sont décrites au moyen de l'Abstract Syntax Notation .1 (ASN.1[+]).

Qu'est ce qu'un *objet*? Un objet est une information sur laquelle portent les opérations du protocole. C'est ainsi que l'objet cible d'une requête de gestion n'est pas, par exemple, un modem ou tout autre appareil, mais c'est un paramètre, un attribut, une information attachée à cet appareil et permettant, en partie, de le caractériser (par exemple : description de l'appareil, temps écoulé depuis la mise en route, nombre de datagrammes transmis). Tous ces objets se trouvent regroupés au sein de la MIB qui est implémentée au sein de cet appareil. Les types d'objets ne sont pas définis au hasard mais sont liés à ce qu'on appelle la définition de la MIB; il existe une définition générale qui est valable pour tous les composants de réseaux et des définitions spécifiques à certains composants produits par certains constructeurs (enterprise specific). Celles-ci définissent d'une manière arborescente tous les types d'objets possibles dans le cadre de la gestion de réseau(x). La MIB est donc une sorte de base de données dont la structure est arborescente et qui contient toutes les informations sur l'appareil et son fonctionnement. Par exemple, la notion de description d'une interface (ifDescr) sera repris dans une définition de MIB comme un type d'objet, mais dans un cas particulier il prendra une certaine valeur (instance) propre à l'interface rencontrée. Il est évident que la MIB d'un modem, par exemple, ne contiendra pas des instances de tous les types d'objets, mais ne comportera que ceux qui sont utiles à sa gestion et ceux qui sont, par définition, obligatoires. On parlera dans ce cas de la *vue de la MIB* (MIB view) attachée au modem. Lorsqu'il nous arrivera de parler de la MIB sans précision, nous parlerons en fait de l'arbre de définition de tous les éléments pouvant se retrouver dans la MIB d'un appareil quelconque. Cette MIB, au sens général du terme, ne contient

[+] ASN.1 = recommandation UIT-T X.208.

bien évidemment que des types d'objets qui seront instanciés par l'implémentation dans un composant du réseau. A ce niveau, il est donc plus juste de parler de type d'objet que d'objet. Nous retiendrons pour l'instant qu'un type d'objet se retrouve dans la définition de la MIB qui est un espace virtuel dont la structure est arborescente. Notons enfin que rien n'est dit sur la façon de référencer des instances d'objets. Chaque protocole déterminera les mécanismes ad-hoc.

Définissons maintenant la **SMI**. Nous avons vu que tous les objets de la MIB font l'objet d'une définition qui utilise un sous-ensemble de la syntaxe ASN.1; ce sous-ensemble est appelé la SMI (Structure of Management Information - Structure d'Information de Gestion). Il s'agit en fait des recommandations portant sur la structure des informations de gestion utilisées pour gérer des réseaux basés sur Internet, le but étant d'arriver à une standardisation. La SMI contient également des macros ASN.1 qui permettent de définir formellement des types construits utilisés dans la définition des types d'objets. Par analogie, on pourrait dire que la SMI est la grammaire et le dictionnaire nécessaires à la description des informations et des types d'objets manipulés. Un type d'objet est défini au moyen d'un NOM, d'une SYNTAXE et d'un ENCODAGE.

Le NOM est utilisé pour identifier un type d'objet et tous les noms sont organisés hiérarchiquement. Indépendamment de la sémantique associée à cet objet, c'est la notion d' IDENTIFIANT D'OBJET (OBJECT IDENTIFIER) qui modélise ce concept, par exemple : 1.3.6.1.2.1.3.1. En deux mots, nous pouvons dire que ce nom est composé d'une suite de chiffres attribués à chaque niveau de l'arbre. Pour des raisons de lisibilité, outre une suite de chiffres (son identifiant d'objet), chaque noeud de l'arbre, sauf la racine, porte une étiquette (un nom ou une abréviation compréhensible). Chaque noeud et chaque feuille voit son identifiant d'objet assimilé à un couple dont le premier élément est l'étiquette du noeud dont il est issu, le second étant le numéro de sa branche (toujours différent de zéro), par exemple : at 1. Aucune limitation n'existe; il est donc possible de redéfinir et d'étendre la MIB sans cesse, à condition de respecter les règles définies dans la SMI. Ainsi une nouvelle version de MIB peut :

- ☑ déclarer d'anciens types d'objets obsolètes mais jamais les détruire,
- ☑ préciser la définition de types d'objets correspondant à des listes en ajoutant des types d'objets non-agrégés aux types d'objets de la liste,
- ☑ définir de nouveaux types d'objets.

Mais elle ne peut :

- ☒ changer la sémantique des objets définis précédemment, sans en changer le nom.

La SYNTAXE est utilisée pour définir la structure des différents types d'objets. Cette définition est faite au moyen d'un sous-ensemble du langage ASN.1. Trois groupes sont utilisés :

- les types primitifs c.-à-d. INTEGER, OCTET STRING, OBJECT IDENTIFIER et NULL
- les types structurés SEQUENCE et SEQUENCE OF très utiles pour définir des tableaux.
- les types définis basés sur les deux autres types et utiles pour définir des types construits propres à des applications.

Ainsi ont été définis par exemple:

- IpAddress = adresse Internet en 32bits,
- Counter = entier toujours positif ne faisant que croître jusqu'à une valeur maximum, fixée ici à 2^{32}-1, et qui au-delà retombe à zéro pour recommencer à s'incrémenter.

L'ENCODAGE des valeurs des instances des types d'objets se fait suivant les règles du Basic Encoding Rules* (BER).

La SMI définit encore la structure des types d'objets qui sont repris dans la MIB. Ainsi, tout type d'objet sera défini en respectant la structure suivante :

- un OBJECT DESCRIPTOR = l'étiquette, nom textuel compréhensible et imprimable. A celui-ci est accolé l'OBJECT IDENTIFIER tel que défini plus haut,
- une syntaxe,

* Standard ISO 8825

- une description ou définition = une définition textuelle de la sémantique attachée à ce type d'objet et devant être respectée par toutes les instances,
- un accès = une des valeurs suivantes: Read Only, Read Write, Write Only non Accessible,
- un statut = soit mandatory, soit optional, soit obsolete.

Par exemple :

atTable { at 1 } ou (1.3.6.1.2.1.3.1)
Syntax : SEQUENCE OF AtEntry
Definition : The address translation table.
Access : read write.
Status : mandatory.

Ces définitions se veulent extensibles; il est possible d'ajouter ultérieurement d'autres champs pour les nouveaux types d'objets.

Cette partie est un résumé du RFC 1155 [ROSE90].

1.1.2. Management Information Base (MIB).

Avant tout, remarquons que le concept de MIB est commun à SNMP et CMIP, mais qu'il a été adapté à l'utilisation concrète qui en est faite. Nous l'avons déjà dit, la MIB est organisée d'une manière arborescente; mais voyons à présent d'un peu plus près de quoi elle se compose. En fait, la MIB de gestion de réseau(x) se trouve elle-même être un sous-arbre d'une structure plus importante, l'arbre d'enregistrement ISO. La Figure 1.1 nous permet d'avoir une vue sur la situation des éléments de la MIB au sein de cet arbre. Ainsi, tout type d'objet appartenant à la MIB verra son identifiant d'objet commencer par la série de chiffres suivante : **1.3.6.1.2.1 .**

On parle de groupes de types d'objets. C'est la base de la conformité c.-à-d. que si la sémantique d'un groupe est applicable au sein d'une implémentation, celle-ci DOIT implémenter tous les objets de ce groupe. Deux raisons justifient la définition de ces groupes :

1. pour assigner les identifiants d'objet,
2. pour permettre de savoir quels types d'objets on doit implémenter pour assurer la gestion des objets gérés (composants logiques et physiques).

Citons à titre d'exemple :

Le groupe *snmp (11)* : il reprend les informations sur l'implémentation de SNMP dans ce système ainsi que les données acquises par l'expérience (sorte de mémoire du fonctionnement de SNMP). Son implémentation est obligatoire pour tous les systèmes qui supportent une entité SNMP. Certaines des instances des types d'objets auront une valeur nulle car SNMP pourra avoir été optimisé en fonction du rôle joué (géré ou gérant). Il y a une instance par entité SNMP; étant donné qu'un même noeud peut être géré et gérant, il y aura dans ce cas deux entités et donc deux instances du même groupe.

Ce groupe contenant 29 types d'objets, nous n'en citerons que quelques-uns :

snmpInPkts (compteur)	=	nombre total de messages délivrés à l'entité SNMP par le service de transport.
snmpOutPkts (compteur)	=	nombre total de messages délivrés par l'entité SNMP au service de transport.
snmpInGetNexts (compteur)	=	nombre total de GetNext PDU acceptés et traités par l'entité SNMP.
snmpOutGetNexts (compteur)	=	nombre total de GetNext PDU générés par l'entité SNMP.

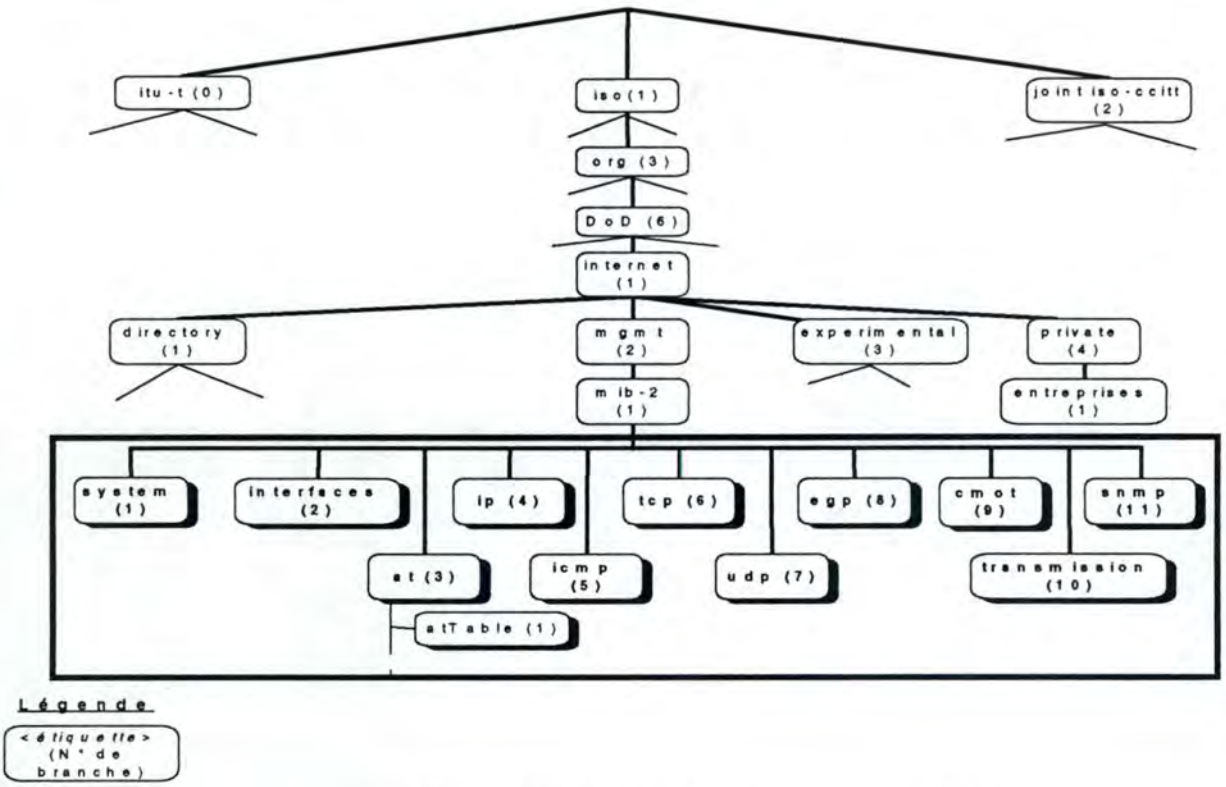


***Figure 1.1** : Positionnement des groupes d'objets de la MIB-II dans l'arbre d'enregistrement de l'ISO.*

Cette partie est un résumé du RFC 1213 [MCCL91].

Terminons en donnant un exemple d'extrait de la MIB d'un routeur IP.

ID = 1.3.6.1.2.1.**1.1.0** .....mib.system.sysDescr.0..... *description de l'appareil.*

Valeur: 3000 software (IGS-KR), version 9.1 (1), SOFTWARE [fcl]. Copyright © 1986-1992 by Cisco Systems, Inc.. Compiled Mon 19-Oct-92 13:52 by jthomas

ID = 1.3.6.1.2.1.**1.2.0** .....mib.system.sysObjectID.0..... *identité du logiciel agent.*

Valeur: object-identifier = .iso.org.dod.internet.private.entreprises.9.1.5.

ID = 1.3.6.1.2.1.**1.3.0** .....mib.system.sysUpTime.0..... *temps écoulé depuis le lancement.*

Valeur: 112942896 timeticks = *13 jours, 1Hr43min48sec*

ID = 1.3.6.1.2.1.**1.4.0** .....mib.system.sysContact.0..... *nom de la personne responsable.*

Valeur: XXXX

ID = 1.3.6.1.2.1.**2.1.0** .....mib.interfaces.ifNumber.0..... *nombre d'interfaces.*

Valeur: 2

ID = 1.3.6.1.2.1.**2.2.1.2.1**.....mib.interfaces.ifTable.ifEntry.ifDescr.1..... *description de l'interface.*

Valeur: Ethernet0

ID = 1.3.6.1.2.1.**2.2.1.2.2**.....mib.interfaces.ifTable.ifEntry.ifDescr.2..... *description de l'interface.*

Valeur: Serial0

ID = 1.3.6.1.2.1.**2.2.1.3.1**.....mib.interfaces.ifTable.ifEntry.ifType.1 ..... *type d'interface.*

Valeur: 6 = *ethernet- csma-cd*

ID = 1.3.6.1.2.1.**2.2.1.5.1**.....mib.interfaces.ifTable.ifEntry.ifSpeed.1..... *vitesse de transmission en bps.*

Valeur: 10 000 000 = *10 Mbps*

ID = 1.3.6.1.2.1.**2.2.1.11.1**.....mib.interfaces.ifTable.ifEntry.ifInUcastPkts.1.. *nombre de paquets unicast transmis vers le haut.*

Valeur: 1379508

ID = 1.3.6.1.2.1.**2.2.1.14.1**.....mib.interfaces.ifTable.ifEntry.ifInErrors.1.... *nombre de paquets écartés pour des erreurs de formatage.*

Valeur: 0

ID = 1.3.6.1.2.1.**4.1.0** .....mib.ip.ipForwarding.0..... *agit comme un hôte ou une passerelle.*

Valeur: 1 = *passerelle*

ID = 1.3.6.1.2.1.**4.3.0** .....mib.ip.ipInReceives.0..... *nombre de datagrammes reçus du bas.*

Valeur: 2774344

ID = 1.3.6.1.2.1.**4.4.0** .....mib.ip.ipInHdrErrors.0..... *nombres de datagrammes écartés pour des erreurs de formatage.*

Valeur: 6

ID = 1.3.6.1.2.1.**4.6.0** .....mib.ip.ipForwDatagrams.0..... *nombres de datagrammes transmis.*

Valeur: 2497661

1.1.3. Simple Network Management Protocol

L'IAB (Internet Activities Board) recommande que toute implémentation basée sur TCP/IP soit gérable au moyen de la SMI, de la MIB et de SNMP. C'est ce dernier qui fera ici l'objet de toute notre attention. Bien que non compatible avec son prédécesseur (SGMP), SNMP en garde la philosophie, l'architecture et l'idée de conception. Nous nous efforcerons d'aborder les points essentiels de ce protocole qui se veut avant tout **simple** et **pratique**.

a. Architecture

SNMP se base sur un modèle architectural des plus élémentaires. En effet, on y retrouve des stations de gestion et des éléments gérés. Ces derniers sont par exemple des hôtes, de passerelles, des serveurs, ... Chaque élément géré implémente sa propre MIB qui contient les divers objets gérés tels que définis ci-dessus. De plus, chaque élément a un agent de gestion qui répond aux requêtes des stations. Les stations de gestion exécutent des applications de gestion qui observent, surveillent et contrôlent les éléments au moyen de requêtes envoyées aux agents. Les stations et les agents communiquent au moyen du protocole SNMP (Figure 1.2).

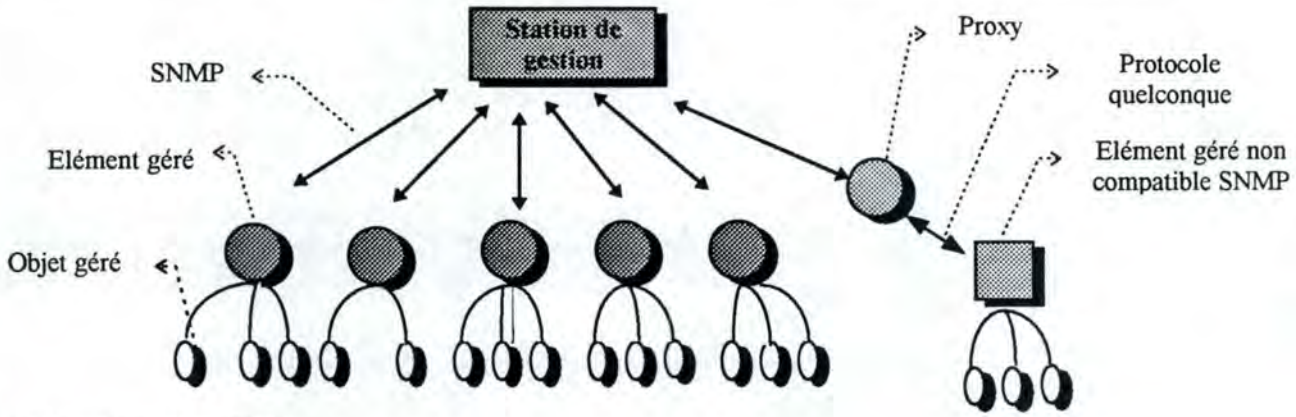


***Figure 1.2** : Architecture SNMP.*

Cette architecture vise à minimiser le nombre ainsi que la complexité des fonctions de gestion implantées dans l'agent. De plus, le paradigme sous-jacent à la surveillance et au contrôle est suffisamment extensible pour permettre des évolutions aisées suite à de nouveaux aspects. Enfin cette architecture se veut indépendante de l'architecture et des mécanismes résidant dans les hôtes et les passerelles.
Les opérations du protocole sont exécutées dans une structure administrative définissant les politiques d'autorisation et d'authentification.

b. Proxy.

Nous pouvons définir un PROXY comme un noeud jouant le rôle d'intermédiaire entre certains éléments du réseau (éventuellement non-accessibles au moyen de SNMP) et le gestionnaire. Lorsque la station veut consulter la valeur d'un objet appartenant à un des éléments couverts par un proxy, c'est à ce dernier qu'elle doit s'adresser. Diverses raisons peuvent être à l'origine de cette solution, par exemple :

- l'élément n'implémente pas la totalité de la suite TCP/IP
- l'élément supporte TCP/IP mais n'implémente pas SNMP ou n'a pas la possibilité d'implémenter la MIB
- on veut limiter l'interaction entre l'élément du réseau et la station de gestion. Dans ce cas, le proxy aura également une politique d'accès associée à l'objet qu'il représente[STAL94].

Outre sa propre MIB, le proxy possède également une MIB décrivant chacun des équipements couverts [NACH95].

c. Concepts.

Les relations entre agents et gestionnaire se font dans un cadre administratif dont les principaux concepts sont détaillés ci-dessous. Les relations entre ceux-ci sont modélisées dans la figure 1.3.

- Les entités qui résident dans la station de gestion et dans l'agent et qui communiquent via SNMP sont appelées les *entités d'application SNMP*. La paire de processus qui implémentent SNMP et donc supportent les entités d'application SNMP sont appelés les *entités de protocole*.
- Chaque agent n'implémente qu'un sous-ensemble de toutes les instances possibles des types d'objets de la MIB. Ce sous-ensemble est appelé une *vue SNMP de la MIB* (MIB View).
- Une *communauté SNMP* (Figure 1.3) est une relation entre un agent et un sous-ensemble donné de stations de gestion. On retrouvera plusieurs stations de gestion pour des raisons de sécurité (Back-up) ou lorsque la gestion est distribuée. Cette communauté se définit par l'authentification, le contrôle d'accès et le service de proxy offert. Notons que ce concept est propre à un agent; ainsi plusieurs agents peuvent avoir le même nom de communauté sans que cela représente un lien quelconque entre ceux-ci. Un agent peut définir plusieurs communautés qui porteront des noms distincts au sein de cet agent. Toute station voulant communiquer avec un agent devra joindre un nom de communauté au message envoyé afin de se faire reconnaître par l'agent. Les stations tiendront la liste des communautés associées aux agents avec lesquels elles peuvent communiquer.
- Un message provenant d'une entité d'application SNMP appartenant à une communauté est appelé par cette communauté *message SNMP authentique*. L'ensemble de règles permettant

d'identifier ce message comme authentique est appelé *schéma d'authentification*. L'implémentation par une fonction d'un tel schéma est appelé *service d'authentification*. Il faut noter que le schéma d'authentification défini par SNMP est des plus rudimentaires puisqu'il s'appuie uniquement sur la reconnaissance de communauté, un peu comme si le nom de communauté était un mot de passe.

- Le *mode d'accès* est représenté par READ-ONLY ou READ-WRITE. L'association d'un mode d'accès et d'une vue est appelée *profil de communauté*. Si on associe un profil de communauté à une communauté, nous avons alors une *politique d'accès* (Figure 1.3) [CASE90]. Cette politique d'accès définit

  - les objets gérés accessibles à une station au sein d'un agent donné
  - les actions (lecture, écriture,...) possibles sur ceux-ci.

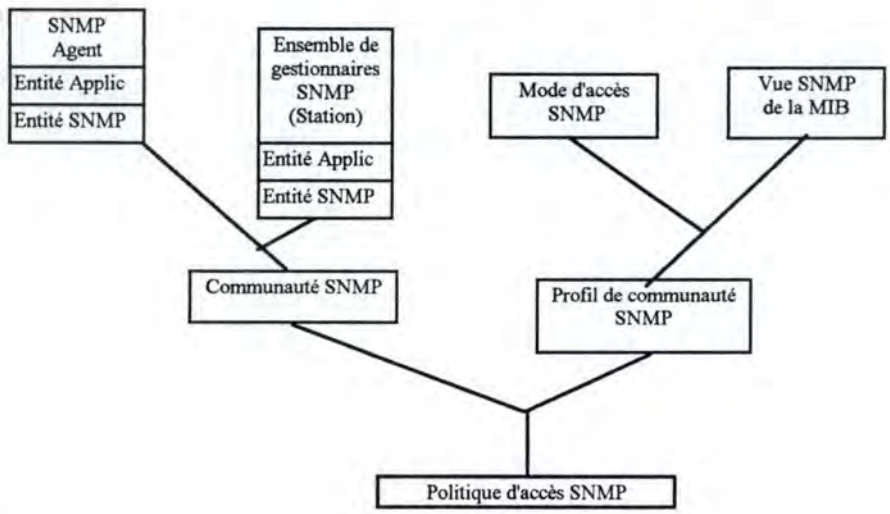


Figure 1.3 :Modèle administratif.

d. Opérations du protocole.

Voyons à présent les opérations supportées par SNMP. Avant tout il faut savoir que SNMP ne permet que de manipuler des instances de types d'objets non agrégés (des tableaux ou des listes). Souvenons-nous que le mécanisme de désignation d'une instance d'un type d'objet de la MIB est du domaine du protocole. SNMP a choisi de désigner une instance d'un type d'objet au moyen d'un identifiant d'objet de la forme **x.0** (pour les types objets scalaires) où x représente l'identifiant d'objet de ce type d'objet dans la MIB. Lorsque l'élément est dans un tableau, la règle générale d'accès veut que son identifiant d'objet soit de la forme **x.y** où x représente l'identifiant d'objet de la colonne contenant les index et y la valeur de l'index de la ligne du tableau. [STAL94].

SNMP prévoit deux types d'interactions :

① Interaction *requête-réponse* où une station envoie une requête à un agent qui lui répond (Polling). Ce type d'interaction est utilisé pour consulter ou modifier des informations de gestion (objets gérés). TROIS services basés sur ce principe existent; ce sont GET, GET-NEXT et SET.

Deux types d'opérations sont possibles:

- Consultation de la valeur d'une variable (instance de type d'objet) avec le GET ou le GET-NEXT.
- Modification de la valeur d'une variable (instance de type d'objet) avec le SET.

② Interaction *non-confirmée* où un agent envoie un message non sollicité (un TRAP) à une station qui ne lui donne aucune réponse. Ce type d'interaction est utilisé pour signaler à une station une situation exceptionnelle qui a produit un changement dans un objet géré de l'agent. Le service de TRAP trouve sa raison d'être dans le fait qu'un agent doit pouvoir prévenir le gestionnaire de l'imminence d'un événement alors que s'il attend la prochaine interrogation il sera trop tard car à ce moment il sera, par exemple, hors-service.

Ces interactions se concrétisent par l'émission de messages entre entités SNMP. Un message "emballe" un PDU. Chaque type d'interaction définit un format de PDU.

Ainsi, une interaction requête-réponse utilisera un PDU reprenant (Figure 1.4) :

- son type,
- un numéro l'identifiant, le RequestID (qui permet la mise en correspondance de la requête et de la réponse),
- un statut et un index d'erreur (qui ne sont utilisés que dans les réponses.),
- un ensemble de couples-variables (variable bindings) composés d'un identifiant d'objet sur lequel porte la requête et sa valeur.

PDU Type	RequestID	Error Status	Error index	Variable-bindings

***Figure 1.4** : PDU d'interaction requête-réponse.*

Get. Ce service permet de consulter les valeurs des objets dont les identifiants sont repris dans la liste des couples-variables. Il est atomique c.-à-d. que s'il n'est pas possible de consulter un objet, aucune valeur d'objet ne sera retournée dans la réponse.

Get-Next. Ce service atomique nous permet d'avoir en retour l'identifiant et la valeur de l'objet suivant lexicographiquement ([1]) chacun de ceux mentionnés dans la liste de la requête. Cela présente un double avantage :

① A moins d'avoir mentionné le plus grand identifiant d'objet ou que la réponse soit de taille trop importante, on aura toujours une réponse. Ainsi, il n'est plus nécessaire que l'identifiant repris dans le couple-variable soit celui d'un objet feuille.

② Cela facilite la consultation de tables. En effet, en mentionnant comme identifiant non plus une "colonne" du tableau suivi d'une valeur d'index mais simplement le nom de la colonne , on obtient le premier élément . En répétant l'opération avec les noms obtenus on peut aisément reconstituer une table.

Set. Ce service permet de modifier la valeur des objets repris dans la liste des couples-variables. Dans ce cas, la liste des couples-variables de la requête comprendra les identifiants d'objets à modifier accompagnés de leur nouvelles valeurs.

Une interaction non-confirmée (Trap) utilisera un PDU reprenant (figure 1.5) :

- son type,
- divers renseignements permettant d'identifier la cause et l'heure du problème ainsi que l'objet concerné,
- l'adresse IP de l'agent ayant causé le TRAP (agent-addr).

PDU type	enterprise	agent-addr	generic-trap	specific-trap	time-stamp	variable-bindings

***Figure 1.5** :Trap PDU.*

e. Protocole de transport.

SNMP a besoin d'un protocole de transport pour échanger ses messages. Toutefois rien n'est dit dans le protocole quant aux services sous-jacents ou quant au fait qu'il soit *connection oriented* ou *connection less.* Bien que l'implémentation la plus courante soit faite sur TCP/IP et utilise le protocole de transport User Datagram Protocol (UDP), il est également possible d'utiliser le protocole Connection Less Transport Protocol (CLTP) de la suite OSI.

(*). Ordre lexicographique : étant données deux séquences d'entiers non-négatifs $(x_1, x_2,...x_n)$ et $(y_1, y_2, ... y_m)$, nous pouvons dire que $(x_1, x_2,...x_n)$ précède $(y_1, y_2, ... y_m)$ dans un ordre lexicographique si la condition suivante est vérifiée : $[(x_j = y_j$ pour tout j tel que $1 <= j < k)$ ET $(x_k < y_k$ pour $k <= n, m)]$ OU $[(x_j = y_j$ pour tout j tel que $1 <= j < k)$ ET $(n < m)]$

f. Bilan

Nous terminerons ce bref tour d'horizon du protocole SNMP avec un bilan des avantages et inconvénients:

- ☺ Il est simple, d'architecture limitée mais efficace.
- ☺ Il repose sur la suite TCP/IP qui est populaire.
- ☺ Son adressage est un adressage IP.
- ☺ Il définit des standards d'objets gérés au moyen des MIB.
- ☺ Il définit des domaines (groupes logiques) avec les notions de vue, de communauté et de profil.

- ☹ Il a des performances insuffisantes pour les gros réseaux.
- ☹ Il est mal adapté à la consultation de grands nombres de données.
- ☹ Les traps ne sont pas confirmés.
- ☹ De sécurité limitée, il est meilleur pour la surveillance que pour le contrôle.
- ☹ Il n'existe pas de manière directe pour lancer une action.
- ☹ La définition de la MIB est limitée et sans sophistication.
- ☹ Il n'y a pas de moyen de communication entre gestionnaires.

Cette balance est issue de [DATS93].

1.1.4. Evolutions de SNMP [STAL94].

Le protocole SNMP a d'emblée connu un franc succès auprès des gestionnaires réseaux. Ceci est vraisemblablement dû à sa simplicité, à son implémentation sur TCP/IP et au fait qu'il n'existait pas vraiment de concurrent. Les nombreux utilisateurs ont fait pression afin qu'il réponde davantage à leurs besoins et aux problèmes qu'ils rencontraient. Une première adaptation a été l'ajout d'un nouveau groupe dans la MIB, le groupe Remote Network Monitoring (RMON MIB). Celui-ci reprend des informations sur tout un sous-réseau (par exemple : un LAN) et non plus sur un seul appareil comme le font les autres groupes de la MIB.

La deuxième évolution notable date de 1992. A cette époque, deux points faibles de SNMP commencent à devenir très critiques: ce sont le manque de sécurité et la relative faiblesse face à de gros réseaux. Deux groupes proposent alors une réponse (Figure 1.6).

Le premier propose une *version sécurisée* appelée Secure-SNMP (S-SNMP). Mais cette version présente le gros défaut de ne pas être totalement compatible avec SNMP.

Le second groupe propose une *modernisation* de SNMP connue sous le nom de Simple Management Protocol (SMP). Cette version améliore la sécurité en reprenant les principes de S-SNMP et étend les fonctionnalités de gestion offertes.

Suite au travail de ces deux groupes, il est décidé de proposer une nouvelle version de SNMP, soit SNMPv2. Deux groupes sont formés. Le premier a en charge le développement des fonctionnalités de SNMP tandis que le second prend en charge l'aspect sécurité, leur base de travail étant SMP. La nouvelle version est terminée en février 93 et proposée comme standard.

Suite à ces développements, les utilisateurs se voient confrontés à de nouveaux choix. En effet, soit ils choisissent SNMPv2 qui n'est pas encore stabilisé, soit ils restent avec la première version en attendant que la nouvelle se stabilise ou que l'approche OSI émerge. A l'heure actuelle, il est difficile de prévoir quelle approche va émerger dans l'avenir. En effet, OSI est fortement soutenu par les organisations d'Etat et supranationales (CEE), alors que les utilisateurs travaillent déjà depuis un certain temps avec des appareils SNMP.

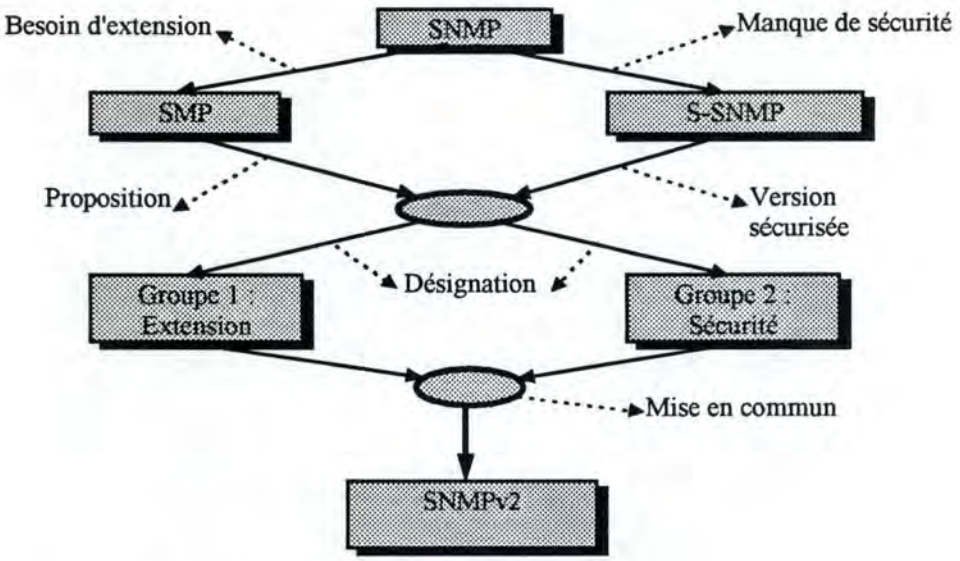


Figure 1.6 : Historique de SNMPv2

1.1.5. RMON.

Les spécifications de RMON [WALD91] sont principalement la définition de modules de la MIB. Il s'agit de définir de nouveaux objets gérés, de nouvelles informations qui portent sur un sous-réseau tout entier. Le but est de définir de nouvelles fonctions de gestion de sous-réseau et d'interfaces entre les stations de gestion et les moniteurs distants (RM = Remote Monitor). Ces derniers peuvent être des appareils dédicacés à cette tâche ou bien des équipements cumulant cette tâche avec leur fonctionnement normal (par exemple : PC sur un LAN). Outre la récolte de toutes les données (ce qui diminue la fréquence de polling de la part de la station) le RM peut effectuer des analyses détaillées des données ne fournissant à la station que les données agrégées en la libérant ainsi de ces traitements. Il est même possible de faire exécuter des processus de diagnostic par le RM ou plus simplement de le charger de l'observation des données afin de détecter certains problèmes. Enfin, précisons encore qu'un même RM peut dépendre de plusieurs stations, ce qui accroît la fiabilité des transmissions d'informations.

1.1.6. SNMPv2

Pour cette partie, outre les RFC, nous nous référons à [STAL94].

Les améliorations de SNMPv2 sont axées sur un besoin de moyens pour gérer des réseaux de taille importante avec plus de sécurité et de précision. SNMPv2 étend la SMI et définit de nouveaux types d'objets. Il présente une solution au problème de sécurité en introduisant les notions de PARTIE et de CONTEXTE. Il offre deux nouveaux services facilitant la gestion de réseau(x) de taille importante (GetBulk - Inform). Nous ne reprendrons ici que les points importants de cette nouvelle version; à aucun moment nous ne prétendons être exhaustif.

a. SMI.

En ce qui concerne les types d'objets qui apparaissent dans cette version, le plus important est probablement la reconnaissance d'une adresse NSAP (OSI). De même, la SMI définit des types de notifications afin de formaliser davantage les messages émis par un agent ou une station vers une autre station. Enfin, elle introduit les *conventions textuelles* (Textual Conventions). Ce sont des types utilisables dans les définitions de modules de la MIB plus lisibles pour les humains.

Disons encore que la manipulation des tables est facilitée et que la procédure de création/destruction de lignes est clarifiée par cette version.

b. Sécurité.

Au point de vue de la sécurité et donc du modèle administratif, SNMPv2 remplace la notion de communauté par celle de PARTIE (PARTY) qui est un héritage de S-SNMP. Ce concept peut se ramener à l'idée de contexte

d'exécution d'une opération. Cette PARTIE diffère suivant le rôle joué et dès lors il se peut que de multiples PARTIES coexistent pour une même entité. Si on veut comparer cette idée avec une communauté, on peut dire qu'il faut deux PARTIES pour remplacer une communauté. [DAVI92] définit une PARTIE comme un contexte d'exécution virtuel, conceptuel, *dont les opérations sont limitées à un sous-ensemble déterminé de toutes les opérations possibles* d'une entité SNMP particulière.

Une PARTIE contient, entre autres, un protocole d'authentification (pour garantir la source et l'intégrité des messages émis) et un protocole de secret (Privacy) contre la divulgation du contenu des messages reçus. Lorsqu' une entité SNMP traite un message, elle le fait en agissant comme une PARTIE et est donc limitée aux opérations définies pour celle-ci. Le mécanisme d'authentification repose sur la mise en place d'un condensé (chiffré) résultant de l'exécution d'un algorithme sur une partie du message. L'aspect secret (Privacy) résulte lui d'un encodage de l'ensemble du message à l'exception de l'identification du destinataire. Lors de l'émission d'un message, une entité pourra choisir d'utiliser ces deux mécanismes, un seul ou aucun de ceux-ci.

SNMPv2 ajoute encore le concept de CONTEXTE qui est défini comme étant une collection d'instances d'objets gérés accessibles par une entité SNMPv2 [GALV93]. Ce contexte peut être accessible soit localement (Local context) soit à distance (Remote context). Si le contexte est local, il est identifié comme une vue MIB et l'accès aux informations de gestion identifiées par le contexte se fait au moyen des mécanismes locaux. S'il est distant, il sera identifié par une relation de proxy au sein de laquelle une entité agent agit comme un proxy pour accéder aux informations de gestion identifiées par le contexte. Dans ce cas, l'interaction a lieu entre le gestionnaire et le proxy tandis que le contrôle d'accès est du domaine de la relation entre le proxy et sa cible. SNMPv2 gagne en souplesse en séparant ainsi les concepts déjà présents dans l'idée de communauté.

c. Opérations du protocole.

Les modifications apportées au protocole sont des extensions de SNMP. En conséquence, une certaine compatibilité est assurée et il est donc logique que le format des messages soit le même. La seule modification se situe dans le "header" qui détermine la politique d'authentification et d'autorisation. Avant de rentrer dans les nouveautés, ajoutons encore qu'un autre type d'interaction est apparu: il s'agit d'une interaction de type requête-réponse où une station gestionnaire envoie un message à une de ses semblables qui lui répond. Ce type d'interaction est utilisé afin de porter une information de gestion à la connaissance d'une autre station.

Un seul format de PDU existe désormais pour tous les types d'interaction. Seul le PDU de la requête de consultation multiple (GetBulk) aura des champs d'une signification particulière.

GetBulk. Il s'agit certainement là de l'amélioration la plus marquante de cette version. Ce service fonctionne comme Get-Next et corrige un manquement de SNMP en permettant la manipulation d'un nombre important de données. Dans le PDU du GetBulkRequest on retrouve deux nouveaux champs venant remplacer les champs de statut et d'index d'erreur qui n'étaient jamais utilisés dans les Get et GetNext-Request (Figure 1.7). Ces champs contiennent chacun une valeur. Le premier, "non-répétées", indique le nombre de variables dont un seul successeur est demandé. Le second, "max-répétitions", donne le nombre de suivants lexicographiques demandés pour toutes les variables ayant un index supérieur à la valeur de non-répétées dans la liste des couples-variables.

PDU type	requestID	non-répétées	max-répétitions	couples-variables

***Figure 1.7** : GetBulk PDU*

Si la taille de la réponse est plus grande que la taille autorisée, la réponse contiendra le maximum de valeurs possibles afin de ne pas dépasser la limite. Si tous les couples-variables n'ont plus de suivant, on n'ira pas voir plus loin et la réponse ne contiendra que les valeurs obtenues.

Inform. C'est la deuxième grande nouveauté de SNMPv2. Ce service est utilisé pour la communication entre gestionnaires (le nouveau type d'interaction). Le PDU est celui utilisé dans les interactions requête-réponse. Les premières variables de la liste des couples-variables sont fixées; elles représentent toujours le moment de l'évènement et l'identifiant d'objet de l'évènement qui peut avoir été défini au moyen d'un type de notification.

d. Protocole de transport.

Avant de parler de la MIB associée à ce protocole, nous aimerions encore signaler que contrairement à SNMP qui ne disait rien sur son implémentation, les spécifications, plus précisément le RFC1449, abordent ici le problème en parlant de son implémentation sur UDP (User Datagram Protocol), OSI CLNS, Novell IPX et AppleTalk.

e. MIB.

Trois nouveaux modules de la MIB sont définis :

- La snmpMIBObjects qui décrit le comportement d'une entité SNMPv2.
- La MIB gestionnaire à gestionnaire (manager-to-manager MIB) qui consiste en un ensemble de types d'objets décrivant le comportement d'une entité SNMPv2 gestionnaire.
- La MIB de Partie (Party MIB): on y retrouve différents objets utiles pour la définition des divers éléments repris dans le modèle administratif.

f. Coexistence.

Nous terminerons en parlant de la coexistence de SNMPv2 avec SNMP. Avant tout, rappelons-nous que cette deuxième version n'est qu'une évolution de la précédente et que ses concepteurs ont tenté de rendre la transition la plus douce possible. La solution passe certainement par des gestionnaires qui peuvent interagir avec des agents des deux types. Les problèmes rencontrés peuvent alors être de deux ordres. Tout d'abord, ***au niveau des informations*** de gestion: étant donné que la SMI n'est qu'une extension de la version précédente, il ne saurait y avoir de problèmes de compatibilité. Ainsi, nous pouvons dire que pour rendre la coexistence possible, il n'est pas nécessaire de procéder à des modifications. Cela n'est plus le cas si l'on désire rendre les informations conformes à la SMI de SNMPv2. Ensuite, ***au niveau des opérations,*** il est nécessaire de procéder selon une des façons suivantes : soit on utilise un agent SNMPv2 comme proxy qui traduit les opérations vers un agent SNMP, soit on utilise un gestionnaire bilingue qui choisit les opérations adaptées au protocole utilisé par l'agent.

1.2. La gestion selon OSI (Open Systems Interconnection)

Dans le monde OSI, on parle de System Management (OSI-SM) plutôt que de Network Management, laissant sous-entendre qu'il est question de gérer des systèmes ouverts*. L'ISO (International Organization for Standardization) a été suivi dans son approche par l' UIT-T ou ITU-T (International Télécommunication Union - Secteur normalisation des télécommunications) qui a consacré à ce sujet la série des standards X.700. Enfin nous tenons à signaler que cette partie s'inspire de [STAL94], sauf mention du contraire.

1.2.1. Concepts.

Les standards peuvent être regroupés en cinq catégories et nous nous proposons de suivre en partie cette structure pour découvrir les concepts sous-jacents à ces standards.

1. Structure de gestion OSI.
2. CMIS (Common Management Information Service) et CMIP (Common Management Information Protocol).
3. Fonctions de gestion de systèmes.
4. Modèles des informations de gestion (MIM).
5. Gestion des couches.

Structure de gestion.

La structure de gestion comprend quatre éléments clés (Figure 1.9) :

- Processus d'application de gestion de systèmes (SMAP): il s'agit du logiciel local à un système qui est responsable de l'exécution des fonctions de gestion de système. Il a accès aux paramètres et possibilités du système. Il peut ainsi gérer tous les aspects du système et se coordonner avec les SMAP des autres systèmes.
- Entité d'application de gestion de systèmes (SMAE): c'est l'entité de niveau application qui est chargée de l'échange des informations avec les autres SMAEs du système. Cet échange se fait en respectant le CMIP.

* Système ouvert (X.200) = la représentation dans le modèle de référence OSI des aspects d'un système ouvert réel, c.-à-d. un ensemble autonome de moyens formant un tout pouvant effectuer des traitements et/ou transfert d'informations et qui, dans ses interactions avec d'autres, remplit les conditions des standards OSI.

- Entité de gestion de couche (LME): dans chaque couche de l'architecture OSI sont implémentées des fonctions permettant de gérer cette couche.
- Base d'informations de gestion (MIB): dans celle-ci sont regroupées les informations de gestion de réseau(x) propres à chaque noeud.

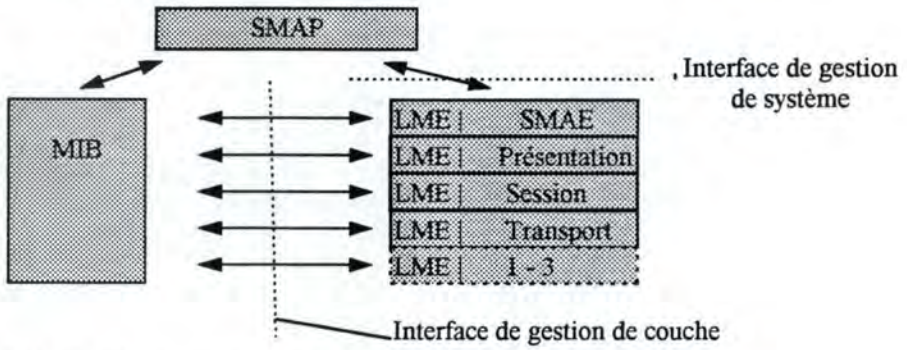
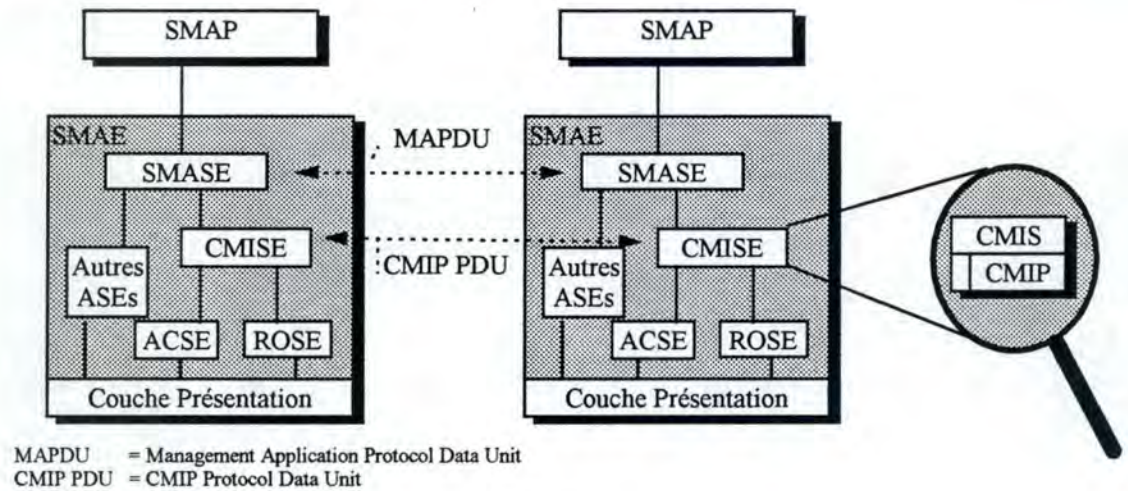


Figure 1.9 : Modèle architectural de la gestion suivant le modèle OSI.



MAPDU = Management Application Protocol Data Unit
CMIP PDU = CMIP Protocol Data Unit

Figure 1.10 : Détail de la couche application en matière de gestion.

Comme toujours dans le modèle OSI, l'entité application, la SMAE (figure 1.10), se décompose en un certain nombre d'éléments (Application Service Element : ASE). Certains sont spécifiques à cette application (par exemple : System Management Application Service Element (SMASE) et le Common Management Information Service Element (CMISE)). D'autres sont communs à de nombreuses applications (par exemple : l'Association Control Service Element - ACSE et le Remote Operation Service Element - ROSE).

SMASE.

Le SMASE reprend les divers services mis à la disposition du gestionnaire et des applications de gestion (par exemple : SMAP). Il implémente l'ensemble des fonctions de gestion. Celles-ci se répartissent en cinq aires fonctionnelles de gestion (Systems-Management Functional Area : SMFA) qui ont été définies par l'ISO. Ce sont :

- Gestion des anomalies (**F**ault Management) : détection et correction d'erreurs.
- Gestion de la configuration (**C**onfiguration Management) : contrôle de la configuration des composants du réseau et des entités dans les couches, ceci afin de répondre aux nouveaux besoins, d'isoler les fautes et d'éviter la congestion.
- Gestion d'informations comptables (**A**ccounting Management) : détermination et attribution des coûts et charges résultant de l'utilisation des ressources réseau.
- Gestion des performances (**P**erformance Management) : évaluation et surveillance des performances du système et des entités dans les couches.

- Gestion de la sécurité (Security Management) : gestion des services utilisés pour fournir la protection d'accès aux ressources.

L'ensemble de ces cinq aires est traditionnellement désigné par "FCAPS". Ces concepts ont été reconnus par tous, que ce soit le monde TCP/IP ou les autres standards propriétaires. Dès lors, cela sort de l'approche purement ISO et UIT de la gestion de réseau(x).

Les aires fonctionnelles sont de grands domaines de responsabilité de la gestion de réseau(x) mais chacune suppose l'emploi d'un certain nombre de fonctions de bases plus ou moins spécialisées; ce sont les fonctions de gestion de systèmes (Systems-Management Functions = SMF). (Figure 1.11). Chacune de ces SMF peut également employer une ou plusieurs SMF afin de remplir sa tâche. Elles sont autant de modules pouvant être utilisés par une ou plusieurs aires.

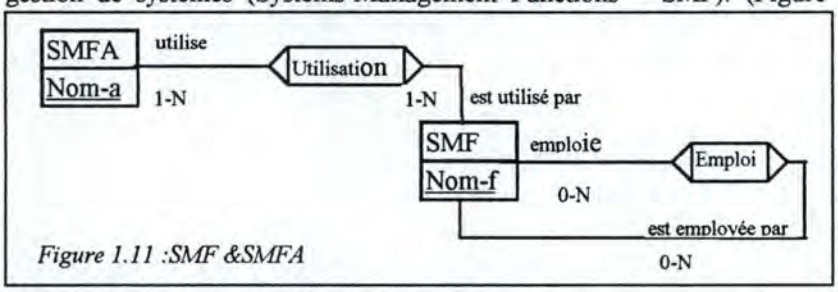


Figure 1.11 :SMF &SMFA

Afin de garder la souplesse qui lui est chère, le monde ISO a préféré définir et standardiser ces fonctions de base plutôt que les SMFA. Treize SMF sont définies; à titre d'exemples nous pouvons citer :

- Gestion d'objet (Object management) : supporte la manipulation d'objets et de leurs attributs et spécifie la notification émise lors des changements de valeur d'un attribut.
- Rapport d'alarme (Alarm reporting) : supporte la définition des alarmes suite aux erreurs et la spécification des notifications émises pour les rapporter.
- Contrôle d'accès (Access contol) : supporte le contrôle des accès aux informations et aux opérations de gestion.
- Compteur de dépenses (Accounting meter) : fournit les moyens de comptabiliser les dépenses (suite à l'utilisation des ressources) et de faire respecter les limitations imposées.
- Surveillance de charge (Workload monitoring) : permet la surveillance des attributs des objets gérés portant sur les performances de la ressource.

Ces SMF reposent à leur tour sur les services de CMISE pour ce qui est de l'échange d'informations.

CMISE.

CMISE fournit quant à lui un ensemble de fonctions de base utilisées pour la gestion de systèmes ainsi que pour la communication inter-SMAE. Il se compose de CMIS qui offre les services aux applications de gestion et de CMIP qui est le protocole utilisé pour échanger les informations provenant de CMIS et repose en fait sur ROSE.

1.2.2. MIB.

Un système comprend un certain nombre d'objets. Chaque objet est en fait une structure de données correspondant à une entité gérée, une ressource matérielle ou logicielle. Cette structure est composée d'attributs, des opérations que l'objet autorise sur lui-même, des notifications pouvant être émises et enfin des relations avec les autres objets. Les seules activités de gestion autorisées dans le monde OSI portent sur les objets plutôt que sur les ressources qu'ils représentent. On peut aisément faire l'analogie avec la définition d'objets faite dans les langages dits orientés objets. Les informations de gestion sont en fait les valeurs des attributs définis dans la structure de représentation, mais le stockage n'est précisé d'aucune manière et c'est donc par une fonction locale de "mapping" que cette information est rendue compatible avec la communication agent - gestionnaire. Notons qu'à ce propos, tout SMAP peut jouer le rôle d'agent ou de gestionnaire. L'ensemble des objets gérés d'un système avec leurs attributs représente la MIB de ce système.

On retrouve également des objets de couche (N) qui sont des ressources spécifiques à une couche (N) et des Objets gérés de système (System managed objects) qui représentent des ressources appartenant à plus d'une couche.

Un objet géré est donc une abstraction directement disponible pour les SMF. D'une manière générale (Figure 1.12), un objet peut représenter une ou plusieurs ressources et inversement, une même ressource peut être représentée par un ou plusieurs objets. Toutes les ressources ne sont pas représentées; on se limitera à représenter les ressources gérées. Enfin, certains objets n'existent que pour la gestion et ne représentent aucune ressource (par exemple : Fichiers d'évènements).

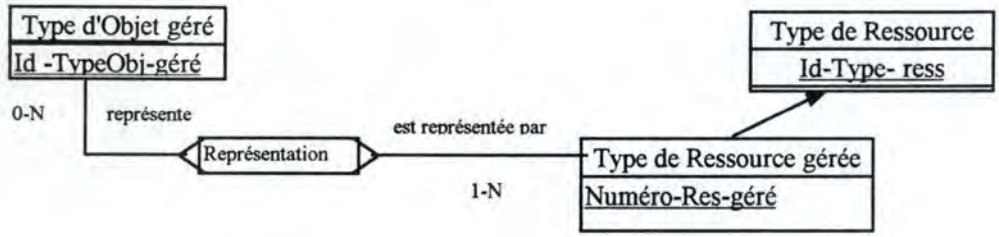


Figure 1.12 : Schéma Entité-Association Ressource - Objet géré

Le standard X.720 (ISO 10165-1) définit le modèle des informations de gestion (Management Information Model - MIM) c.-à-d., entre autres, la SMI et les objets gérés. Ainsi pour structurer la définition de la MIB, chaque objet est une instance d'une classe d'objet (ou type d'objet). Celle-ci est un "patron" pour les objets gérés partageant la même définition. Chaque type de ressource gérée est une classe d'objet dont une instance est un objet géré. Les données et les procédures applicables à ce type de ressource sont encapsulées dans le type d'objet correspondant. Enfin, toutes les opérations sur un objet géré se font au moyen d'un "message" adressé à cet objet. Cela permet de garantir l'intégrité des objets.

Une classe d'objet géré se définit par :

- un ou des attributs visibles,
- une ou des opérations de gestion de système qui lui sont applicables,
- un ou des comportements en réponse aux opérations,
- des notifications pouvant être émises,
- des ensembles conditionnels pouvant être encapsulés dans l'objet géré,
- la position de l'objet géré dans la hiérarchie d'héritage.

On retrouve ici encore une certaine analogie avec l'approche orientée objet. La conception orientée objet permet l'ajout de fonctions ou de classes d'une manière modulaire. Il n'est pas nécessaire d'avoir recours à un système de gestion de base de données orienté objet pour l'implémentation. On se limite ici à l'utilisation des principes de l'orienté objet. A ce propos, nous allons en voir quelques-uns utilisés dans la définition des objets gérés.

Attribut (variable*).

L'attribut est l'élément de donnée d'un objet géré (Figure 1.13). Il reflète une propriété de la ressource que l'objet représente (par exemple : caractéristiques opérationnelles). Il est habituellement utilisé pour la surveillance mais aussi le contrôle. Il est bien entendu typé mais possède également une règle d'accès (READ, WRITE, READ-WRITE) et des règles de correspondance permettant de le localiser au moyen d'une sélection, d'un filtrage (matching rules). Il peut être scalaire ou ensembliste (SET-OF). Enfin, on peut grouper plusieurs attributs, ce qui permet d'agir sur l'ensemble de ceux-ci avec une seule opération.

Opération (message*).

Une opération peut s'appliquer soit à un attribut ou groupe d'attributs, soit à l'objet entier. Le système qui invoque cette opération doit avoir les droits nécessaires.

Comportement (méthode*).

Le comportement d'un objet géré est une réponse à un stimulus externe (opération de gestion de système = message CMIP) ou interne (événement interne à l'objet géré et à sa ressource associée).

* Terme utilisé dans l'approche orientée objet classique.

Notification (message*).

Un objet géré émet une notification lorsqu'il détecte une occurrence interne ou externe l'affectant. Elle est soit émise soit conservée.

Ensemble conditionnel (inexistant dans l'approche Orientée objet classique).

Un ensemble conditionnel est un groupe d'attributs, de notifications, d'opérations et de comportements qui sont optionnels mais liés (tous ou aucun). La condition de leur présence est généralement le reflet d'une possibilité de la ressource. Ce n'est pas une construction indépendante de l'objet; elle fera partie de l'héritage.

Héritage (héritage).

Cela permet de définir une nouvelle classe d'objet sur base d'une ou plusieurs classes existantes. La nouvelle classe est alors appelée sous-classe. Cette technique a deux conséquences :

- ☞ Elle permet de développer une hiérarchie qui modélise bien la structure des ressources.
- ☞ Elle oblige la sous-classe à hériter de TOUTES les caractéristiques de sa mère (super-classe).

Bien que l'héritage multiple soit permis, il est difficilement utilisé.

Allomorphisme (polymorphisme*).

L'allomorphisme permet d'avoir des objets différents qui présentent la même interface avec le système; cela permet également à une sous-classe d'émuler le comportement de sa super-classe. Certaines restrictions doivent être faites quant aux conditions d'héritage. Ainsi, en ce qui concerne les super-classes et la portée des attributs de la sous-classe, le domaine de valeur des attributs de la sous-classe doit être inclus dans celui de la super-classe. La relation d'allomorphisme se caractérise par un attribut ensembliste (SET-OF) reprenant l'ensemble des super-classes imitées.

Contenance (contenance*) et dénomination.

Nous venons de parler de l'héritage, mais dans la gestion de réseau(x) suivant l'ISO, il faut distinguer la hiérarchie d'héritage qui définit la relation entre classes d'objets et qui par conséquent facilite la définition et la conception de nouveaux objets, et la hiérarchie de contenance qui affecte les instances au sein de la MIB. Ainsi, certains objets sont composés d'un certain nombre d'objets de niveau inférieur (Répertoire, fichier, champs) mais un objet est composant d'au plus un objet supérieur (Figure 1.14).

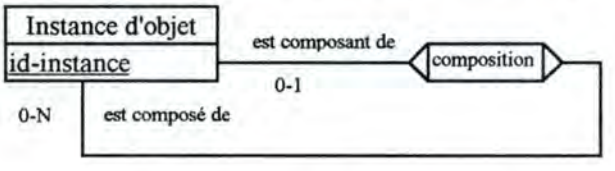


Figure 1.14 : Modèle Entité-Association exprimant la hiérarchie de contenance.

Au point de vue des noms, il faut une fois encore distinguer les classes des instances (contrairement à SNMP où le nom d'instance était directement issu de la classe). Ici le nom d'instance est obtenu comme suit :

- Chaque classe d'objet géré possède un attribut qui est utilisé dans le nom de ses instances.
- Le nom distinct relatif (relative distinguished name) d'une instance est une valeur pour cet attribut. Il est unique parmi tous les subordonnés du même supérieur.
- Le nom distinct d'une instance est représenté par la séquence des noms distincts relatifs depuis la racine jusqu'à cet objet.

Une autre façon serait de dire qu'il existe TROIS structures d'arbre.

1. L'arbre d'enregistrement ISO qui est aussi celui utilisé par TCP/IP (pour les définitions des noms et pour la MIB). Ici, il ne sert qu'à fournir le nom des classes d'objets gérés, la définition des attributs, des actions, notifications et ensembles conditionnels. C'est en fait un dictionnaire ou une boîte de mécano où se trouvent toutes les pièces nécessaires à la construction de nouveaux objets.
2. L'arbre d'héritage qui indique comment une définition de classe d'objet est dérivée d'une autre.

3. L'arbre de contenance qui représente en fait la structure de la MIB montrant les objets contenus et la hiérarchie d'objets.

Outre les définitions des objets, le MIM (Management Information Model[+]) spécifie également les opérations possibles sur ceux-ci. Deux catégories sont à distinguer. Tout d'abord, les opérations sur les attributs (Get valeur d'attribut, Replace valeur d'attribut, Set to default, Add et Remove member); celles-ci sont similaires à celles de SNMP puisqu'elles peuvent porter sur un ensemble d'attributs. Par défaut l'opération est non atomique mais il est possible de demander l'atomicité. Si elle n'est pas atomique, la réponse comportera la liste des attributs et leur valeur associée qui, en cas d'erreur, se verra remplacée par un code erreur. Le second type d'opération porte sur les objets (Create, Delete, Action). La sémantique de ces opérations fait partie de la définition de la classe d'objet géré.

Le standard X.721 (ISO 10165-2) définit quelques classes d'objets, des attributs et des notifications pouvant servir de base au développement d'une MIB. Parmi les types d'attributs, certains types génériques ont été définis; c'est notamment le cas des jauges, compteurs qui bien que portant le même nom qu'en SNMP sont d'utilisation plus souple. A titre d'illustration, citons la distinction des compteurs paramétrables ou non suivant le fait qu'on puisse leur affecter une valeur ou non, ou bien encore l'attribut de "tidemark" pouvant être affecté à une jauge et représentant, par exemple, la valeur maximum depuis un certain temps. On retrouve également des attributs spécifiques qui sont quant à eux complètement définis au moyen du langage ASN.1 et qui se voient attribuer un identifiant d'objet. Sont encore définis des types de notifications pouvant être appliqués à une large variété de classe d'objet. Ils sont définis au moyen des formats des données de notification, de son comportement, du format des données de résultat et d'un identifiant d'objet.

Terminons ce point en attirant l'attention une fois encore sur le fait que rien n'est dit sur l'implémentation de la MIB. Sachant qu'un équipement minimal d'un tel système se compose d'une station de gestion et d'un certain nombre de ressources, il est possible de stocker l'information de la MIB à un seul ou à plusieurs endroits, avec ou sans duplication. Si une duplication est choisie par souci de sécurité, il faudra alors veiller à la cohérence des données ainsi stockées.

1.2.3. Common Management Information Service (CMIS)

Si un système de gestion repose sur une base de données, sa fonction primaire est avant tout l'échange d'informations (de gestion) entre 2 entités (SMAE) au moyen d'un protocole. C'est ce que fait CMISE. Comme nous l'avons déjà dit lorsque nous avons vu les concepts, CMISE se compose en fait de CMIS et de CMIP. CMISE repose également sur ACSE.

CMIS est l'ensemble des services utilisables par les SMF de SMASE. Il offre trois types de service regroupant les sept services offerts par CMISE ainsi que ceux qu'il ne fait que relayer:

- Service d'association (Association service) : avant de communiquer, les utilisateurs de CMISE doivent créer une association. CMIS ne fait que relayer la demande de service à ACSE qui se charge de cette tâche.
- Service de gestion des notifications (Management-notification service) : ce service est utilisé pour le transport des informations de gestion des notifications.

1	M-EVENT-REPORT	confirmé / non-confirmé

- Service des opérations de gestion (Management-operation service) : ce service est utilisé pour le transport des informations de gestion des opérations de gestion de systèmes.

2	M-GET	confirmé
3	M-SET	confirmé / non-confirmé
4	M-ACTION	confirmé / non-confirmé
5	M- DELETE	confirmé
6	M-CREATE	confirmé
7	M-CANCEL-GET	confirmé

CMIS offre deux facilités pour les services 2-3-4-5:

[+] Le MIM est le nom donné à l'ensemble des définitions de la SMI et de la MIB.

- Possibilité de réponses multiples à une opération confirmée en les liant à celle-ci. Cela se fait au moyen de l'identifiant de liaison (équivalant à l'identifiant d'invocation) qui est présent dans chaque sous-réponse. Ce mécanisme est semblable à celui du request-id (identifiant de requête) de SNMP.

- Possibilité de faire porter des opérations sur des objets gérés multiples, sélectionnés sur un certain critère et sujets à certaines conditions de synchronisation.

De plus, trois outils permettent de procéder à une sélection de un ou plusieurs objets. Il y a tout d'abord la PORTEE (Scoping): il s'agit en fait d'identifier un ou plusieurs objets gérés qui représenteront le domaine auquel va être appliqué un filtre. Partant de l'arbre de contenance et ayant choisi un objet géré appelé *objet géré de base* (OGB), on choisira parmi ses successeurs (ses composants) ceux auxquels on appliquera ultérieurement un filtre. Ce pourra être l'OGB seul, ses composants du niveau N, l'objet géré de base et tous ses composants jusqu'au niveau N ou enfin l'objet géré de base et tous ses composants. Le deuxième outil est justement le FILTRAGE (filtering): il consiste en fait en une expression booléenne contenant une ou plusieurs assertions quant à la présence ou la valeur d'attributs dans un objet géré retenu après application de la portée. Chaque assertion sera testée pour l'égalité, l'ordonnancement, la présence ou la comparaison. Le troisième outil, la SYNCHRONISATION, est utile lorsque la portée et le filtrage ont sélectionné plusieurs objets gérés. La question qui se pose est alors de savoir comment gérer ces multiples objets. Deux choix sont possibles : *Atomique* ou *Best-effort* (tous les objets effectuent l'opération ou la tentent).

Nous allons à présent décrire les services offerts par CMIS en commençant par le service de notification de gestion.

- **M-EVENT-REPORT**. Ce service est fort semblable au *Trap* de SNMP. Comme lui, il est issu de l'agent et est destiné au gestionnaire. Il est confirmé ou non. Chaque notification est accompagnée d'un numéro permettant de l'identifier. Si une erreur s'est glissée dans la notification et si le service est confirmé, la réponse portera un paramètre d'erreur. Un certain nombre de paramètres d'erreur ont été définis. Ils sont beaucoup plus explicites que ceux présents dans SNMP.

Voyons à présent les services d'opérations de gestion.

- **M-GET**. Ce service permet d'extraire des données de gestion contenues dans la MIB. L'appel du service est issu du gestionnaire. La requête peut porter sur un ou plusieurs objets gérés et sur un, quelques-uns ou tous les attributs qu'ils contiennent. La sélection des objets concernés se fait au moyen des outils définis ci-dessus. Lorsque plusieurs objets résultent de la sélection, la réponse est en fait une réponse multiple. En plus de ses attributs et de leur valeur respective, chaque sous-réponse comporte un identifiant de liaison, sauf la dernière qui ne porte qu'un identifiant d'invocation équivalant à l'identifiant de la requête initiale. Toute erreur entraîne la fourniture d'un code d'erreur. Toutefois, il faut remarquer qu'une erreur sur un attribut de la liste n'entraîne pas l'arrêt de l'opération comme dans SNMP.

- **M-SET**. Ce service a, lui aussi, la même fonction que dans SNMP mais il est beaucoup plus riche. Tout d'abord, il n'est pas nécessairement confirmé. S'il l'est, son fonctionnement est fort similaire à celui de M-GET; néanmoins, certains éléments de M-SET interagissent et rendent ce service plus complexe. Tout d'abord, il y a la portée et le filtrage qui permettent de sélectionner un ou plusieurs objets gérés; tous ne doivent pas avoir le même ensemble d'attributs. Ensuite, il y a le choix de la synchronisation *Atomique* ou *Best-effort*. Enfin, le cas d'un succès partiel est possible, c.-à-d. qu'on peut modifier la valeur de certains attributs au sein d'un objet, alors que ce n'est pas possible pour tous. Ainsi, si l'on désire faire un SET uniquement sur les objets gérés contenant tous les attributs sans qu'il soit nécessaire que cette opération réussisse sur tous les objets gérés, on fera un SET avec la synchronisation de type Best-effort et un filtre utilisant l'assertion "present" pour chaque attribut listé. Enfin, si on procède à une comparaison avec le SET de SNMP, malgré le fait que ce ne soit pas vraiment possible puisque la MIB est différente, on peut d'emblée dire que CMIS offre une plus grande souplesse de sélection des objets gérés. De plus, l'atomicité n'est pas semblable; en effet, les objets manipulés par SNMP sont assimilables à des attributs de OSI-SM. L'atomicité est ici définie au niveau des objets et dès lors, si on veut faire un SET sur divers attributs d'un seul objet géré avec OSI-SM, celui-ci le fera toujours, même s'il n'est pas possible de modifier certains de ces attributs.

- **M-ACTION**. Ce service permet l'appel d'une procédure prédéfinie dans un objet géré. Le request spécifie l'action et les paramètres de celle-ci. Ce service peut ne pas être confirmé. S'il est confirmé, la réponse contiendra la classe et l'instance d'objet géré afin de confirmer l'action. Elle peut éventuellement reprendre le type d'action et les paramètres de réponse donnant des informations sur l'exécution.

- **M-CREATE**. Ce service est utilisé pour créer une nouvelle instance d'une classe d'objet. Il est toujours confirmé. On inclura les éventuels paquets conditionnels et les diverses valeurs des attributs composant cet objet dans la requête. Une autre solution consiste à désigner une autre instance comme modèle. La réponse comprendra toujours l'instance d'objet géré ainsi créée. Elle pourra comprendre la classe d'objet et la liste d'attributs qui reprendra les différents attributs de l'objet et les valeurs de l'instance.

  Avant d'aller plus loin, revenons un instant sur la manière de créer une nouvelle instance et de lui assigner des valeurs au moyen de la liste d'attributs et/ou d'un modèle. Il existe trois techniques de définition d'instances et trois techniques d'affectation de valeurs. Voyons d'abord les procédés de création :

  1. Si le paramètre d'instance supérieure est présent, il identifie dans la MIB cible l'instance devant être la supérieure de la nouvelle.
  2. Si le paramètre d'instance supérieure n'est pas présent, alors l'identifiant d'instance doit être présent.
  3. Si ni le paramètre d'instance supérieure ni l'identifiant d'instance ne sont présents, alors c'est l'utilisateur du CMIS cible qui assigne la valeur à l'identifiant d'instance.

  Pour ce qui est de l'affectation des valeurs, le procédé est quelque peu plus complexe:

  1. Si le paramètre *liste d'attributs* est présent, alors il contient une liste d'identifiants d'attributs ainsi que leur valeur.
  2. Si le paramètre *instance d'objet* de référence est présent, alors il spécifie une instance d'objet de la même classe que le nouveau. Les attributs non repris dans la liste prennent la valeur de ceux présents dans l'objet de référence.
  3. Si le paramètre *instance d'objet* de référence n'est pas présent, alors les attributs absents de la liste d'attributs et pour lesquels une valeur par défaut est définie prennent cette valeur.

- **M-DELETE** est le service inverse de M-CREATE puisqu'il sert à détruire un ou plusieurs objets de la MIB. Il est toujours confirmé et fonctionne comme un GET lorsqu'il y a des réponses multiples.
- **M-CANCEL-GET**. Ce service permet d'arrêter un GET trop long. Il est toujours confirmé. Il est logique que ce type de service ne serve que pour le GET, puisqu'il est le seul service n'affectant pas le contenu de la MIB. Comme toujours, en cas d'erreur, la réponse contiendra un code erreur. La requête ne contient que deux paramètres : l'identifiant d'invocation et l'identifiant d'invocation du Get annulé.

Terminons cet aperçu des services de CMIS par les services d'associations. Ces services sont transparents pour CMIS; les demandes de services émises à destination de CMISE par les SMASE sont en fait des demandes de service ACSE qui sont passées en l'état par CMIS. Rappelons enfin que, pour que deux entités d'application puissent communiquer dans un modèle OSI, il est nécessaire d'établir une association entre celles-ci. Etant donné que cet ASE est général, nous nous contenterons de citer ce qui le caractérise dans le cadre de la gestion de systèmes. Tout d'abord, il y a l'établissement d'une association entre les entités d'applications; ensuite cet ASE permet aux deux utilisateurs de négocier les caractéristiques de CMISE utilisées dans cette association. Ces caractéristiques sont reprises dans des unités fonctionnelles signalant les possibilités de service. Ainsi, tous les services de CMIS sont fournis par défaut à l'exception de M-CANCEL-GET et avec la restriction pour GET, SET et ACTION de ne pouvoir faire des requêtes multiples sont définis. Les trois outils de sélection ainsi que la possibilité de réponses multiples sont définies dans d'autres unités fonctionnelles pouvant faire l'objet de négociations.

1.2.4. Common Management Information Protocol (CMIP) et Remote Operation Service Element (ROSE).

CMIP

CMIP définit les procédures pour l'échange d'informations au moyen de CMIP-PDU entre les CMISE pour l'exécution de services CMIS. Nous avons vu que CMIS offre sept services; certains peuvent ne pas être confirmés et d'autres permettent des opérations multiples. Les services d'associations étant directement traités par ACSE, ils ne sont pas passés à CMIP. Par conséquent, il existe onze types de PDU. Un mapping est fait pour transformer une demande de service CMIS en un type de CMIP-PDU. Chacun de ces PDU peut contenir

jusque trois types d'information: des arguments (request/indication), des résultats et des erreurs (response/confirm). CMIP utilise à son tour ROSE qui est un ASE général (non spécifique à la gestion de systèmes).

ROSE

Le service de base offert par ROSE est l'invocation d'une opération sur un système ouvert distant. Cet ASE est l'un des plus utilisés. Il vise à permettre des applications interactives, c.-à-d. qu'une entité d'application demande à une autre qu'elle exécute une opération particulière. Ce concept est en fait très proche du Remote Procedure Call. L'échange entre les deux entités d'applications se fait dans un contexte d'association d'applications. L'interaction est encore caractérisée par une classe d'opérations négociée par les deux entités d'application pour cette invocation. Ainsi cinq classes d'opérations ont été prédéfinies et se caractérisent par le comportement de compte-rendu et par l'échange synchrone ou asynchrone (Figure 1.15).

		Opérations synchrones	*Opérations asynchrones*
Mode de rapport	Si l'opération réussit, alors la cible renvoie le résultat; si l'opération échoue, alors renvoie l'erreur.	Classe 1	Classe 2
	Si l'opération réussit, alors pas de réponse; si l'opération échoue, alors la cible renvoie le code erreur associé.		Classe 3
	Si l'opération réussit, alors la cible renvoie le résultat; si l'opération échoue, alors pas de réponse.		Classe 4
	Si l'opération réussit, alors pas de réponse; si l'opération échoue alors pas de réponse.		Classe 5

***Figure 1.15** : Classes d'opération de ROSE.*

Nous avons vu que l'interaction se déroule dans le cadre d'une association; aussi faut-il ici distinguer trois classes.

1. Association de classe 1 : seul l'initiateur de l'association peut invoquer une opération.
2. Association de classe 2 : seul le répondant de l'association peut invoquer une opération.
3. Association de classe 3 : tous deux peuvent invoquer une opération. Il est ainsi permis de créer un ensemble d'opérations liées où il y a un parent et 1 à N enfants.

Le fonctionnement de ROSE peut être décrit d'une manière générique comme suit (Figure 1.16): un utilisateur d'un ASE fait appel indirectement au ROSE au moyen d'un OPS.request dont certains paramètres sont utilisés par ROSE, alors que la plupart sont employés par l'ASE, afin de créer un APDU (Application PDU).

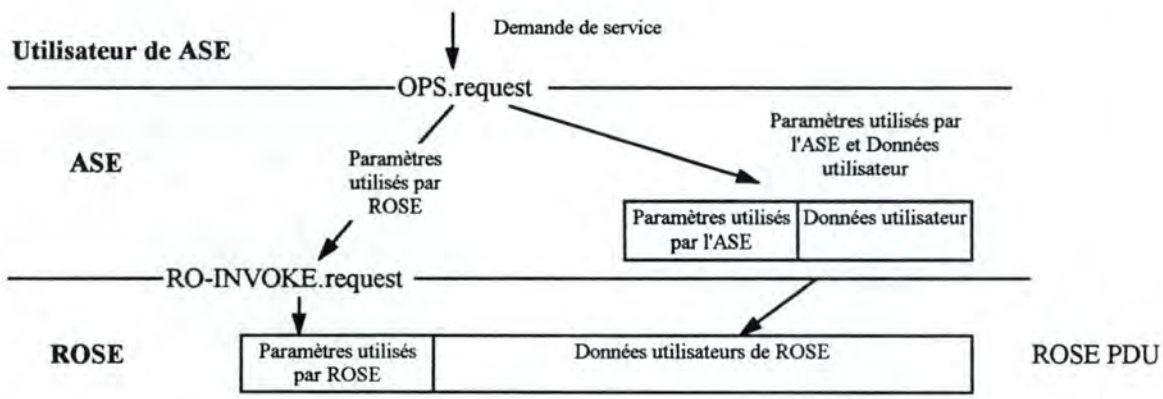


***Figure 1.16** : Utilisation des services de ROSE.*

CMIP utilise des associations de classe 3. Les services de CMIS confirmés nécessitent l'utilisation d'opérations de classe 1 ou 2. Les services de CMIS non confirmés emploient des opérations de classe 5.

1.2.5. Conclusions et remarques.

Remarques (X.700 - ISO 7498-4).

La gestion de couche devrait permettre d'assurer une gestion des éléments intermédiaires tels que les routeurs et bridges, pour autant qu'ils implémentent ces mécanismes de gestion de couche dans les couches qui les constituent. De plus, ces mécanismes permettent que même lors de défaillances de l'application de gestion, on puisse malgré tout bénéficier de moyens de gestion minimum. D'où une meilleure robustesse et une économie des ressources réseau employées. Il faut noter que cette gestion de couche utilisera d'autres protocoles que CMIP. Ces protocoles ne seront utilisés que pour répondre à des besoins particuliers ou lorsque des protocoles de gestion de systèmes sont indisponibles. La sémantique des informations de gestion de couche ainsi que les opérations possibles sur celles-ci doivent rester cohérentes avec ce qui est défini dans la gestion de systèmes.

Pour pouvoir disposer d'une SMAE, il est nécessaire que chaque couche du modèle OSI dispose des fonctions suffisantes pour prendre en charge une telle entité. Si cela n'est pas possible, le niveau maximal de gestion se limitera à l'utilisation des fonctions individuelles assurées par les éléments de la gestion de couche (N) de ce système. Si ni la gestion de systèmes, ni la gestion de couche n'est possible, le niveau maximal de gestion sera alors l'ensemble des fonctions de gestion individuelles assurées par l'exploitation de la couche (N).

Enfin remarquons encore qu'une SMAE peut exister indépendamment de l'existence d'entités de gestion de couche dans une quelconque de ses couches.

Mise en garde.

Un système de gestion de réseau(x) doit veiller à :

- Ne pas dégrader les performances du réseau qu'il est censé maintenir. [STAL94], citant Arondoff & al., parle d'une charge de 5% de la largeur de bande comme étant un trafic maximum.
- Permettre une prise de décision et une action de contrôle rapides, c.-à-d. avant que les conditions du réseau ne changent. Ce qui implique un échange régulier d'informations.
- Fournir un riche ensemble de services afin de manipuler un large domaine de fonctions de gestion et de fournir une surveillance et un contrôle détaillé (génération d'un nombre important d'informations de gestion).

Dans le monde de la gestion suivant le modèle OSI, ces trois points peuvent devenir critiques car :

- OSI-SM offre beaucoup de services et fonctions, ce qui engendre beaucoup de trafic.
- La MIB est de taille importante et complexe.
- Les CMIP PDU sont imposants car ils utilisent l'encodage BER et passent par les sept couches.

Sécurité.

Nous avons vu que parmi les SMF, il y en avait un qui s'appelait Access-Control et qui supporte le contrôle des accès aux informations et aux opérations de gestion. Il spécifie les objets gérés et les attributs à utiliser pour autoriser ou retirer l'accès; ceci, en accord avec la politique de contrôle d'accès représentée par l'information de gestion de contrôle d'accès.

On y retrouve trois mécanismes de contrôle d'accès :

1. *Liste de contrôle d'accès*: elle représente l'ensemble des initiateurs (entité capable d'accéder à une cible) et leurs droits sur la cible (chose dont l'accès est contrôlé) sur laquelle est stockée cette liste.

2. *Ticket de capacité*: il représente pour l'utilisateur auquel il est attaché, les droits qu'il a sur l'ensemble des cibles.

3. *Etiquette de sécurité*: c'est un mécanisme se basant sur le fait que chaque cible est affectée d'un niveau de sécurité et que chaque initiateur s'en voit également doté. Tout initiateur peut lire une cible dont le niveau de sécurité est inférieur ou égal au sien et il peut écrire sur une cible dont le degré de sécurité est supérieur ou égal au sien.

Enfin il faut encore savoir que trois classes d'objets ont été définies spécialement pour représenter les informations de gestion du contrôle d'accès.

Conclusions.

Comme nous l'avions fait pour SNMP, nous terminerons cette partie en reprenant les conclusions de [DATS93].

- ☺ OSI-SM définit les objets gérés en classes d'objets qui ont des attributs, des opérations et des notifications.
- ☺ OSI-SM est supporté par des profils d'applications standardisés en combinaison avec beaucoup de profils de transport.
- 😐 OSI-SM a une architecture complexe mais complète.
- ☹ OSI-SM requiert l'implémentation d'une pile OSI complète.
- 😐 OSI-SM repose sur un adressage de la couche application du modèle OSI.

1.3. Les standards propriétaires.

Jusqu'à présent, nous avons étudié le standard international normalisé par l'ITU-T (OSI-SM et CMIS/CMIP) ainsi que le standard de fait, créé sur l'initiative du monde TCP/IP (SNMP). Ce dernier est déjà une conséquence de la demande des utilisateurs, pressant le monde des communications pour qu'il crée un outil performant de gestion. Outre TCP/IP, il y a un autre monde fort prisé par les utilisateurs, c'est celui d'IBM et SNA. De plus, actuellement, le monde des LAN se développe très fort et un des OS (Operating System) les plus prisés est probablement NETWARE de NOVELL. Nous allons dès lors rapidement passer en revue ce qui est prévu dans ces mondes.

1.3.1. IBM

IBM a développé une architecture devant permettre la gestion de gros réseaux SNA appelée ONA (Open Network-Management Architecture). En publiant le format de ses interfaces et de ses messages, IBM a tenté d'imposer aux vendeurs son architecture comme un standard. La structure de base se compose de trois éléments. Ce sont le *Focal point* (point de focalisation), l'*Entry point* (point d'entrée) et le *Service point* (point de service) (Figure 1.17). D'entrée de jeu, notons que cette architecture est fort proche de celle de SNMP avec laquelle nous ne manquerons pas de faire un parallèle .

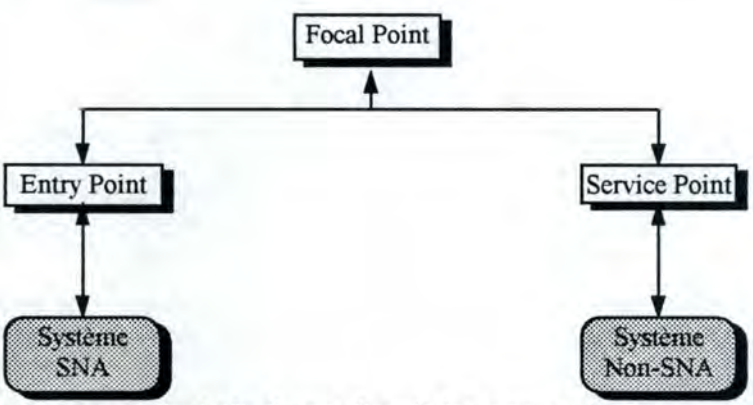


Figure 1.17 : IBM ONA

- Le *Focal point* est l'endroit où sont fournis les moyens d'action aux opérateurs. A cet endroit l'ensemble des informations de gestion est analysé, collecté et stocké. IBM a développé un produit spécifique pour cette fonction: c'est NetView. Nous en reparlerons dans le chapitre 3 (≈Station de gestion).

- L'*Entry point* est un appareil compatible avec SNA qui implémente les services de gestion pour sa propre gestion et/ou pour celle d'autres produits qui lui sont liés (≈ Element manager ou agent). La plupart des produits IBM de réseaux de données peuvent fonctionner comme *Entry point*.

- Le *Service point* est un appareil compatible avec SNA qui implémente les services de gestion pour sa propre gestion et qui rend possible la surveillance et le contrôle par un *Focal point* d'appareils ou réseaux ne pouvant être des *Entry point*s, par exemple, parce qu'ils ne sont pas compatibles avec SNA (≈ Proxy). C'est donc une sorte de passerelle pour la gestion de réseau(x).

Etant donné la pression de la concurrence et la volonté d'assurer une gestion globale des ressources de communication à partir de son *Focal point*, IBM a développé trois nouveaux éléments permettant une gestion à une plus grande échelle (Figure 1.18).

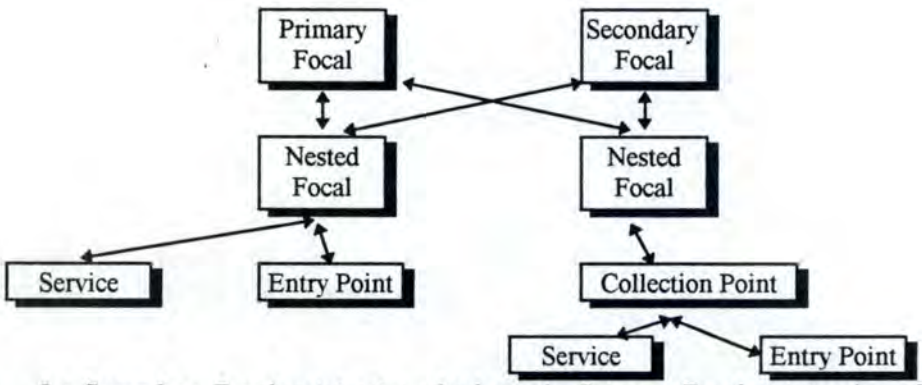


Figure 1.18 : SNA

- Le *Secondary Focal point* est un back-up du *Primary Focal point* prévu afin d'assurer une disponibilité accrue du *Focal point*.
- Le *Nested Focal Point* est un élément qui supervise une partie du réseau lorsque celui-ci est de taille importante (≈ Remote MONitoring - RMON).
- Le *Collection Point* permet d'assurer le relais entre un sous-réseau SNA et un *Focal point* (≈ RMON).

L'intention d'IBM est claire: il s'agit de permettre une gestion de réseau(x) distribuée au sein d'une entreprise. L'implémentation des fonctions de gestion est prévue au niveau des couches supérieures du modèle SNA c.-à-d. comme pour les couches présentation et application. La MIB est construite suivant la structure décrite par le monde OSI. Cette architecture ne signifie pas qu'IBM ne cherche pas à être compatible avec SNMP ou CMISE. Au contraire, Netview est aussi utilisé pour communiquer avec des agents SNMP. De même, NetView peut accéder à des agents CMISE, au moyen d'une logical unit, en émulant un gestionnaire compatible.[TERP92]

1.3.2. NOVELL.

NOVELL n'ayant pas la puissance d'IBM n'a pas cherché à imposer son standard. Conscients du fait qu'ils n'étaient pas les premiers, les concepteurs de NOVELL NETWARE ont plutôt cherché la compatibilités avec les produits et les standards existants. Ainsi ont-ils développé différents modules (NLM - NetWare Loadable Module, VLM - Virtual Loadable Module) venant s'additionner aux produits de base. Dans sa version 4.0 NetWare est devenu totalement compatible avec TCP/IP grâce au module ODI (Open Data-Link Interface). D'autres modules assurent la compatibilité avec SNA, AIX (Unix chez IBM), etc... .

Mais Novell a également développé un logiciel permettant la gestion de réseau(x) au sens large du terme; il s'agit de NMS (NetWare Management System). Celui est un système ouvert, permettant de contrôler et de surveiller des réseaux hétérogènes en se basant sur SNMP. Il faut savoir que ce gestionnaire intègre divers produits de chez NOVELL* et des produits de tiers tels que Intel (LANDesk Manager) et Synoptics (Optivity). Ainsi NMS 2.0 permet de gérer directement:

- ☑ Les serveurs NetWare 3.x et 4.x,
- ☑ Les agents NetWare LANalyser pour l'analyse distribuée de réseaux,
- ☑ Tout appareil compatible avec SNMP,
- ☑ Les routeurs,
- ☑ Les hubs compatibles avec Hub Management Interface,

* Par exemple : des caractéristiques de gestion de serveurs précédemment dans le module Netware Services Manager, des fonctions de gestion de hubs reprises de Hub Services Manager et de nouvelles possibilités de gestion dans LANalyser.

☑ Les adresses réseau.

Il permet la gestion des erreurs, la gestion de la configuration, l'enregistrement de données pour l'analyse et ultérieurement le planning, et la gestion de réseau(x) avec une interface basée sur MS-Windows. De plus, il facilite l'intégration d'applications de gestion développées par des clients ou des tiers [NOVE93]. Ce logiciel est déjà fortement orienté vers une intégration de la gestion des réseaux; c'est pourquoi nous en reparlerons plus tard.

Conscient de l'importance de la gestion des réseaux pour ses clients, Novell a constamment développé des outils facilitant la tâche des gestionnaires. En 1993, Novell a choisi la stratégie NMDS (NetWare Distributed Management Services) s'appuyant sur NMS afin de gérer des environnements distribués. La spécificité de l'approche réside dans le fait que contrairement aux autres plates-formes de gestion où le système de gestion est résident, NMDS est distribué sur le réseau et non sur un appareil; ceci devrait permettre de passer outre des problèmes de congestion pouvant résulter de l'approche classique. Ainsi, la console n'est plus qu'un endroit pour visualiser et non plus un endroit de stockage et d'analyse des informations. Les principes qui lui sont sous-jacents sont :

- un ensemble commun d'informations pouvant être accédé par tous les services,
- une vue unique permettant l'utilisation d'une seule console pour remplir les diverses tâches de gestion,
- des services de gestion tels que gestion des équipements, administration de réseau,... .[CHIP94]

1.4. Autres standards.

1.4.1. OSF/DME.

L'Open Software Foundation's Distributed Management Environment (environnement distribué de gestion) est conçu pour unifier la gestion de réseaux et de systèmes dans des environnements distribués multi-vendeurs. Cet environnement distribué de gestion est profitable à tous:

- il réduit la complexité, la formation et les coûts pour les administrateurs;
- il étend le marché des applications de gestion;
- il accroît la polyvalence des opérateurs en leur permettant de passer de la gestion d'un petit système indépendant à la gestion d'importants réseaux et de super-ordinateurs.

DME assure la consistence dans l'environnement géré en dotant les développeurs d'un environnement de gestion complet. Il permet l'interopérabilité avec la plupart des standards existants et peut être utilisé pour gérer des ressources existantes ou s'intégrer dans d'autres systèmes de gestion.

Ce standard semble actuellement être au point mort et ne fait l'objet d'aucune implémentation; ses principes restent des lignes directrices pour bon nombre de constructeurs.

Pour plus de détails nous renvoyons le lecteur à l'annexe D.

1.4.2. OMNIPoint.

OMNIPoint est un ensemble de recommandations du Network Management Forum (association de vendeurs et d'utilisateurs). Ce programme vise à permettre aux utilisateurs d'exprimer leur demande tout au long de la chaîne au moyen d'un langage compréhensible par les fournisseurs. OMNIPoint fournit la structure pour réaliser une gestion de réseaux et de systèmes basée sur des services. De plus, il permet de sélectionner et d'intégrer les standards et les technologies nécessaires à la réalisation d'une solution réaliste.

L'effort principal de OMNIPoint vise à supprimer les barrières de l'implantation en :

- sélectionnant les standards et spécifications qui seront suivis pour répondre aux besoins spécifiques;
- ajoutant des détails d'implantation aux standards de base retenus;
- intégrant les standards et les technologies sélectionnés, et en développant les spécifications de conversion entre les technologies où cela est nécessaire;
- relevant les besoins d'un environnement pour supporter une telle implantation;
- regroupant les nombreux standards et spécifications en fonction des interactions entre clients et vendeurs.

OMNIPoint vise essentiellement à introduire de la stabilité dans un monde où émergent sans cesse de nouveaux standards. Il fixe ainsi des points de convergence au moyen d'un ensemble de recommandations (par exemple: OMNIPoint 1) et spécifie la stratégie à suivre pour passer d'une version à la suivante.

1.4.3. M.3010.

La recommandation M.3010 définit les principes à suivre pour l'établissement d'un réseau de gestion des télécommunications. Un tel réseau doit permettre aux Administrations de télécommunications d'assurer la gestion de leurs moyens et services. Il permettra ainsi la planification, la mise en oeuvre, l'installation, la maintenance, l'exploitation et l'administration des réseaux et services de télécommunication. Il va de soi qu'elle doit permettre la prise en compte d'une grande diversité de matériel, services et réseaux.

L'architecture d'un tel réseau peut être vue sous trois angles : architecture fonctionnelle, informationnelle et physique. Elle est composée de modules et en permet la distribution parmi les systèmes de gestion. Un exemple de répartition en cinq niveaux est proposée en annexe de cette recommandation.

Pour plus de détails nous renvoyons le lecteur à l'annexe B.

Chapitre 2 : Système intégré de Gestion de Réseaux (SIGR).

2.0. Introduction

Comme le dit fort justement H.A.Wiersinga dans [WIER93], il y a vingt ans, les données, les utilisateurs et les applications se situaient tous au même endroit. Ensuite est venu le temps où les utilisateurs se sont dispersés (hôtes avec terminaux). Après quoi, ce sont les applications qui en ont fait de même (PC groupés en LAN) et bien vite, en raison de l'environnement client/serveur, les données ont suivi le mouvement. Ainsi le nombre de réseaux n'a-t-il cessé de croître ces dernières années. Leur complexité et les services qu'ils offrent ont évolué au rythme des progrès technologiques, tant et si bien qu'aujourd'hui, les utilisateurs sont confrontés à des problèmes de gestion d'un environnement composé de ces divers réseaux et services. Ces réseaux ont acquis au sein des entreprises et de la société un statut privilégié qui les rend stratégiquement et opérationnellement très importants. Désormais, il est devenu nécessaire de planifier leur développement, de les surveiller et de les piloter avec beaucoup de maîtrise. Malheureusement, les outils classiques actuels ne considèrent souvent que les équipements d'un seul constructeur ou au mieux ceux qui sont nécessaires à l'établissement d'un sous-réseau (par exemple : LAN).

L'idée qui commence à apparaître est celle d'une gestion unique, c.-à-d. faire disparaître la multiplicité des réseaux derrière un système de gestion cohérent et complet. L'opérateur pilote une unité logique qui reprend l'ensemble des ressources de télécommunications de l'entreprise. On parlera alors d'intégration ou encore d'unification (cfr. Infra). Comme nous le verrons au point 2.3.3., il est possible d'utiliser SNMP et des Proxies pour réaliser cette intégration (par exemple : SunNet Manager - Cfr Infra).

En informatique, les significations données à l'intégration sont nombreuses mais les définitions sont rares; aussi en avons nous retenues deux. La première est issue de la physiologie, tandis que la seconde est celle donnée par le monde du management (économique).

> L'intégration est la coordination des activités de plusieurs organes, en vue d'un fonctionnement harmonieux, réalisée par divers centres nerveux.[LARO59]
>
> L'intégration est l'art de gérer des éléments dotés de capacités d'autonomie de plus en plus grandes en leur donnant un cadre et un sens de plus en plus affirmés. [MERL95]

2.1. Cadre de référence

Nous avons déjà beaucoup parlé de la gestion de réseau(x) mais n'en avons encore examiné qu'un seul aspect. On peut en fait étudier la gestion d'un réseau sous trois angles différents. On peut comme dans [WIER93] considérer un cube de la gestion où chacune des dimensions représente une façon d'aborder le problème (Figure 2.1).

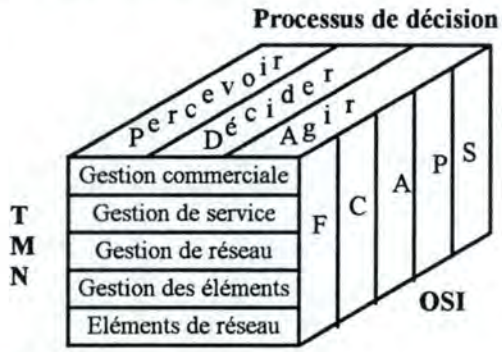


***Figure 2.1** : Cube de la gestion.*

La première dimension concerne les différentes phases d'un processus de décision. En effet, la gestion de réseau(x) est principalement une activité de décision au sens général du terme. Ces phases sont aussi celles que Mintzberg avance dans son modèle du processus décisionnel. Rappelons-les :

- perception (Awareness) : prise de conscience d'une situation,
- décision (Décision) : analyse et détermination des actions à prendre,
- exécution (Implementation) : exécution des actions choisies.

La deuxième dimension est celle proposée par l'UIT-T dans sa recommandation X.720. C'est la dimension fonctionnelle que nous avons jusqu'ici déjà citée. Elle se compose de cinq sections : la gestion d'anomalies, la gestion de la configuration, la gestion de la comptabilité, la gestion des performances et la gestion de la sécurité (cfr. Supra).

Enfin la troisième dimension est celle de l'architecture fonctionnelle définie par l'UIT-T dans son standard M.3010 sur le réseau de gestion des télécommunications (TMN - Télécommunications Management Network). Ce standard distingue cinq couches qui sont à considérer comme autant de niveaux d'abstraction. Ces couches sont :

Les éléments de réseau (Network Element) : en ce qui concerne la gestion de réseau(x), il est certain qu'à ce niveau, il y a peu de choses si ce n'est la génération des alarmes, la tenue de compteurs et d'une base de données interne.

La gestion des éléments de réseau (Network Element Management) : il s'agit ici de systèmes pouvant gérer des éléments spécifiques à un constructeur. Ces systèmes remplissent aussi une fonction de passerelle permettant l'interaction de la couche de gestion de réseau(x) avec les éléments de réseau. Ceux-ci peuvent, par exemple, gérer des modems. Les outils retrouvés à ce niveau sont les superviseurs: par exemple, le système CMS400 de chez RACAL.

Gestion de réseau(x) (Network Management) : cette partie reprend la gestion de tous les éléments (indépendamment du vendeur) composant un réseau, la gestion dite end-to-end. Ce niveau fournit la fonctionnalité nécessaire pour gérer un réseau en coordonnant l'activité sur l'étendue du réseau.

Gestion des services (Service Management) : ici c'est la gestion des services offerts par le réseau qui est visée. Nous devons donc, à ce niveau, considérer également ce qui est appelé la qualité du service (QoS -Quality of Service) fixée dans l'accord portant sur le niveau du service (SLA - Service Level Agreement).

Gestion commerciale (Business Management) : cette couche est du niveau de l'entreprise. Elle s'intègre dans la gestion de l'ensemble de l'entreprise et a de nombreuses interactions avec d'autres systèmes de gestion. C'est le point de convergence de l'action lorsque celle-ci est nécessaire au niveau de l'entreprise.

Remarquons qu'il n'est pas nécessaire pour une système de gestion de remplir toutes les fonctions du cube. Il remplira au moins la/les fonction(s) d'un niveau du TMN et sera en interaction avec d'autres systèmes assurant les fonctions des couches inférieures.

Ce travail porte essentiellement sur des produits se situant au niveau de la couche gestion de réseau(x) mais INTEGREE, c-à-d. supportant des ressources et réseaux divers. Il s'agirait en quelque sorte d'une couche 3' pouvant avoir sa propre couche de gestion des services "intégrés". Nous verrons que certains produits du niveau gestion des services et gestion commerciale seront abordés. Cela se justifie aisément par le fait qu'une entreprise offre, en général, une gamme de produits dont un des composants est du niveau gestion de réseau(x).

2.2. Hétérogénéité.

Malgré les nombreux protocoles et recommandations, l'hétérogénéité est fréquente dans le monde des télécommunications. Tout d'abord, il y a celle dont nous avons le plus entendu parler, c'est la diversité des vendeurs au sein d'un même réseau (y compris au sens Internet du terme) c.-à-d. au niveau des éléments du réseau. Ainsi, la réinitialisation de modems ne se fera pas nécessairement de la même façon d'un fabricant à l'autre. Comme nous l'avons sous-entendu ci-dessus, ce sera le gestionnaire d'éléments (superviseur), si on en dispose, qui gommera ces différences. Cette diversité a causé la multiplication des gestionnaires de réseaux classiques dans les salles de contrôles actuelles.

Ensuite, au niveau de la gestion des réseaux, il existe une certaine hétérogénéité qui trouve ici son origine dans des protocoles différents, des représentations de données différentes, des noms différents, des réseaux de données différents, ... Il suffit de repenser aux protocoles de gestion vus au chapitre 1.

Enfin il y a l'hétérogénéité des domaines, des types de réseaux et de systèmes (LAN, MAN, WAN, Operating System ,...) au sein de l'entreprise. Leur intégration accrue accentue les problèmes issus de cette hétérogénéité alors que jusqu'à présent ils étaient peu perçus.

Ainsi les opérateurs sont-ils confrontés à une multitude de types d'informations, de commandes et procédures. Cela rend leur tâche plus ardue voire impossible. D'où des temps de réaction aux incidents croissants et un niveau de service ne correspondant plus à celui négocié avec les clients (internes ou externes). Deux solutions sont possibles: soit augmenter le nombre d'opérateurs, soit trouver un moyen technique d'effacer cette différence. On comprendra aisément que, à long terme, la solution technique risque d'être économiquement

plus viable. C'est ainsi qu'est né le concept d'intégrateur et c'est donc autour de l'hétérogénéité que s'est élaboré le concept de SIGR.

Dans certains cas, hormis la réduction de personnel en salle de contrôle, un intégrateur se justifie car, même en l'absence de connexion physique, le comportement d'un réseau peut avoir des répercussions sur un autre. Il est donc utile d'avoir une seule vue de ces deux réseaux. Pour illustrer ce que nous venons de dire prenons l'exemple d'une société qui offre deux réseaux à son personnel pour communiquer par courrier électronique ou pour le transfert de fichiers. Si l'un des deux vient à voir ses performances chuter pour une raison technique ou autre, on peut fort logiquement craindre que les utilisateurs se reporteront sur le second dont les performances s'en ressentiront; ils ne sont pourtant pas physiquement reliés.

Ainsi, nous pensons qu'une certaine corrélation doit être faite avec des réserves, entre des sous-réseaux interconnectés en un seul réseau hétérogène et entre des réseaux non-interconnectés ayant un lien logique entre eux. Le grand problème qui reste dans ces deux cas, comme dans celui où il n'existe aucun lien entre les réseaux gérés (par exemple LAN et PABX), est de masquer l'hétérogénéité du/des réseaux, notamment au moyen d'une approche orientée objet. Suivant la fonction attendue du système de gestion, il nous semble qu'il faille distinguer l'intégration de l'unification. En effet, s'il existe un lien entre les divers réseaux, on parlera fort logiquement d'*intégration* puisqu'on les ramène à un seul grand réseau, tout au moins au niveau de la gestion. Mais si on veut administrer divers sous-réseaux indépendants au moyen d'un même outil qui les présente suivant la même logique, il nous semble plus juste de parler d'*unification*.

En conclusion de l'article [BHUS94], on relève les conseils suivants pour la conception d'un intégrateur: il faut

- ☑ représenter les composants d'un réseau par des classes d'objets,
- ☑ représenter les paramètres statistiques d'un réseau sous forme d'attributs d'une classe d'objets,
- ☑ spécifier le type d'attributs,
- ☑ effacer l'hétérogénéité au moyen d'agents,
- ☑ identifier les points d'accès des constructeurs (localisation et extraction de l'information),
- ☑ construire un niveau d'abstraction (via les niveaux sémantiques et syntaxiques).

Le principe de base utilisé pour effacer l'hétérogénéité au niveau des informations, c'est *l'unification*. Nous en reparlerons dans le Pt 2.3.2 lorsqu'il sera question de la passerelle de gestion.

2.3. Système Intégré de Gestion de réseau(x) (SIGR).

2.3.1. Définition d'un SIGR.

[BHUS94] définit un SIGR comme *un système d'applications permettant à un utilisateur final d'intégrer, de contrôler, de gérer des réseaux hétérogènes comportant une multitude de produits, de traitements et de communications.*

Objectif.

Le but d'un SIGR est d'offrir à l'opérateur, grâce à une seule interface, un ensemble unique d'informations et d'outils d'administration lui permettant d'avoir une meilleure préhension des problèmes et diminuant ainsi le niveau d'expertise requis pour être efficace. De plus, il permet l'intégration de modules lui apportant une plus-value, par exemple des modules d'intelligence artificielle et un système expert pouvant contribuer à l'analyse des évènements.

2.3.2. Architecture générique [TERP92].

Nous nous proposons d'expliquer les divers concepts importants d'un SIGR en partant de l'architecture générique (Figure 2-3) proposée par [TERP92] dont l'idée de base est orientée modèle OSI; en outre, on peut également y remarquer les concepts théoriques de la recommandation M.3010 de IUT-T (voir Ann.). Nous pouvons retrouver l'idée de SMF (System Management Function) dans les éléments d'application et l'idée de SMFA (System Management Functional Area) dans les applications de gestion intégrée de réseaux orientées tâche. Nous allons à présent détailler les divers composants de cette architecture et les comparer avec ce qui existe dans les standards.

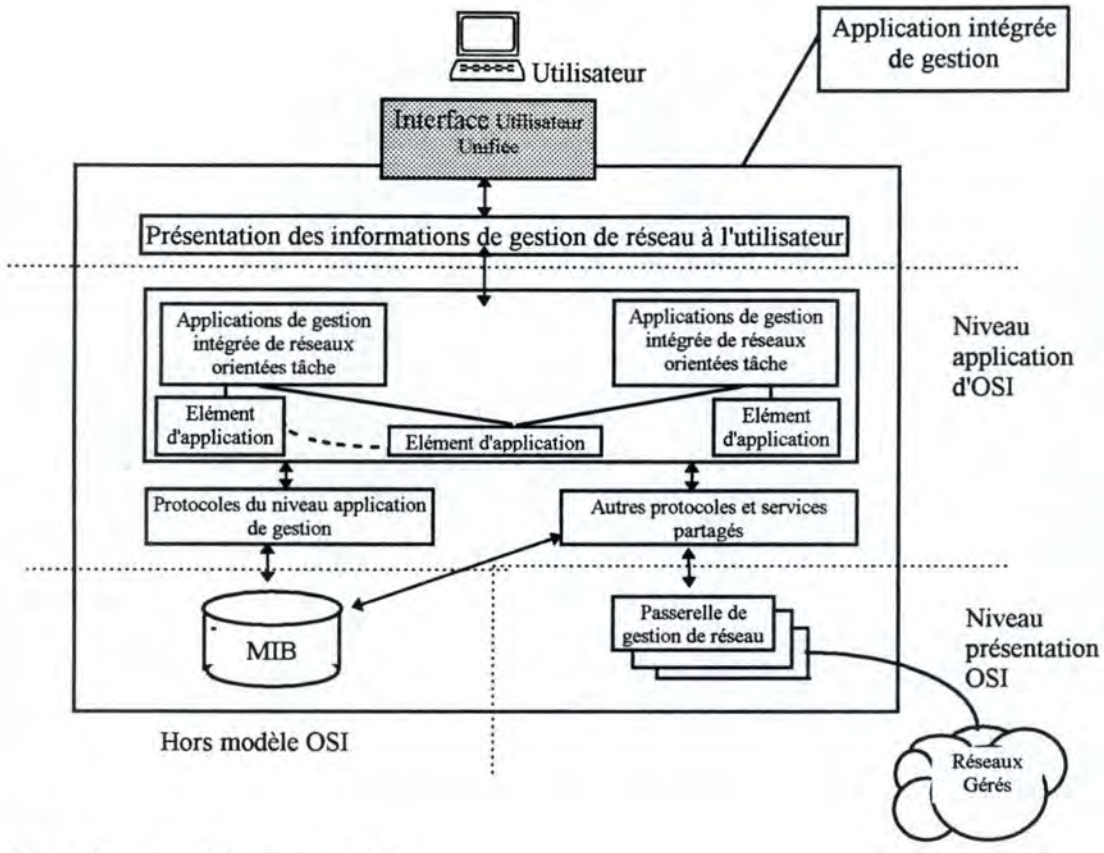


Figure 2-3 : Architecture générique d'un SIGR.

Interface utilisateur unifiée.

C'est le moyen d'interaction entre l'opérateur et le SIGR. Elle se caractérise par :

- un point de passage unique pour la gestion des différents réseaux,
- des notations symboliques des objets et données gérées,
- des possibilités de zoom permettant d'avoir une vue hiérarchique du réseau,
- un multifenêtrage et un environnement multitâches,
- un système d'aide et de conseil intégré.

Le but de cette interface est de faciliter le travail de l'opérateur en diminuant son adaptation aux systèmes de gestion (Cfr. Pt 2.5.2.). On peut penser que l'emploi de concepts génériques peut faciliter le passage à l'automatisation du pilotage des résaux. Il semble qu'à l'heure actuelle il n'en soit rien, car la conception, la corrélation des événements complexes et les choix nécessitent toujours une décision humaine; cependant l'intelligence artificielle peut aider les décideurs (Cfr. Pt 2.5.3).

Service de présentation.

Afin de fournir une image unique des réseaux, il faut tenir compte des conventions sémantiques et syntaxiques en matière d'informations destinées à l'opérateur. Le service de présentation doit permettre la traduction des informations d'entrée/sortie. Avec l'interface unifiée, il vise à offrir à l'utilisateur une représentation des réseaux la plus fidèle et la plus explicite possible.

Les éléments d'application et les applications orientées tâches.

Comme nous l'avons dit ci-dessus, on retrouve dans cette architecture la philosophie des SMF et SMFA. La grande différence par rapport à ces dernières (implantées dans les systèmes natifs), c'est qu'ici, les éléments d'application et les applications orientées tâches se situent au niveau du SIGR et lui fournissent des fonctions indépendamment du réseau sur lequel porte la fonction invoquée. Elles donneront à leur tour accès aux systèmes natifs ad-hoc. Pour l'utilisateur, le comportement du SIGR est toujours le même, à savoir générique. Les éléments d'application représentent le niveau le plus bas du système; ils sont comparables à des modules dans un logiciel orienté objet. Ils offrent des services aux applications de gestion orientées tâches en utilisant

éventuellement d'autres éléments d'application. C'est également à leur niveau que se situent ou peuvent se situer des composants d'intelligence artificielle (Cfr. Pt 2.5.3.). Les applications fournissent à leur tour des services à l'utilisateur en s'appuyant sur un ou plusieurs éléments d'applications. Ces services sont les seuls perçus par l'utilisateur.

MIB ou repository.

La MIB, tout comme les éléments d'application et les applications orientées tâche, n'est pas à confondre avec ce qui se trouve au niveau des systèmes natifs. En effet, le SIGR repose sur l'idée d'une vue unique et générique des réseaux hétérogènes. Les objets de gestion associés seront par conséquent des objets génériques représentant les composants gérés. Il faut donc distinguer les RMO (Real Managed Objects - Objets Gérés Réels) des GMO (Generic Managed Object - Objets Gérés Génériques). Les RMO sont les composants logiques et physiques d'un réseau (par exemple : appareil d'Entrée/Sortie, ports, qualité de service). Ils pourront être décrits suivant leur structure, leur comportement et leur pilotage. Les applications intégrées de gestion de réseau(x) ne peuvent agir sur ces objets qu'au moyen des GMO et de passerelles de gestion. Les GMO sont les représentations génériques des RMO; c'est la vue informationnelle qu'en a l'application intégrée de gestion. Ils peuvent être vus comme étant composés de ressources OSI (génériques par définition) et de ressources réelles génériques (appareil d'Entrée/Sortie générique, routeur générique,...). Ils sont décrits en termes de structure, de comportement et de pilotage génériques. Il faut remarquer que l'article [ABEC93] fait référence à une telle approche.

La MIB doit donc ici être vue comme un "repository " conceptuel des occurrences d'informations de gestion génériques, dont des sous-ensembles décrivent des GMO et des occurrences d'informations de gestion génériques reflétant les besoins particuliers des applications intégrées de gestion (par exemple : attribut d'un "trouble ticket* " pour une application de gestion d'anomalies). Les applications seront conçues sur base des GMO qui représenteront d'une manière unique et consistante les RMO du système. [TERP92]

Protocoles.

Un des éléments clés d'une approche intégrée est la fourniture d'une organisation distribuée, flexible et ouverte de l'application intégrée de gestion. Cette flexibilité signifie qu'il doit être possible de grouper des éléments d'information génériques (= partition de la MIB) et des éléments d'application (= gestionnaire de réseau). Aucun de ces ensembles ne peut être assigné d'une manière restrictive au système à partir duquel ils seront distribués. La coopération se fera au moyen d'un protocole du niveau application. L'ouverture quant à elle signifie qu'une interopérabilité doit être assurée entre les offres des divers vendeurs grâce à l'utilisation de protocoles de gestion de réseau(x) standards; à ces protocoles s'ajoutent divers services utilisés pour la distribution du système de gestion.

Passerelles de gestion de réseau(x).

Il s'agit là de l'élément primordial du gestionnaire intégré. En effet, outre l'offre d'une présentation unique à l'opérateur, l'intégrateur doit être compatible avec les réseaux gérés et avec les divers protocoles de gestion utilisés par ceux-ci. Cela n'est possible que par l'entremise de passerelles de gestion de réseau(x). Celles-ci vont traduire les RMO en GMO, isolant ainsi l'opérateur des systèmes natifs. De même, les actions génériques requises par les applications intégrées seront traduites par des actions des applications natives qui interagiront avec les objets natifs du réseau géré (Figure 2-4).

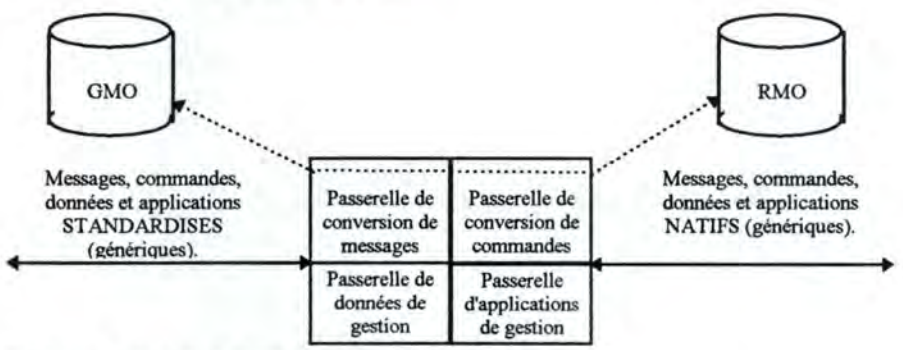


Figure 2-4 : Rôle d'une passerelle de gestion.

* Trouble ticket = enregistrement conservé dans une base de données qui reprend divers renseignements concernant un problème qui est survenu. Sa création peut être automatique ou manuelle par un opérateur d'un "help desk". On pourra y ajouter toute information utile pour une résolution ultérieure (par exemple: action exécutée). Lors de la confrontation à un problème, on pourra rechercher dans la base de données tous les problèmes similaires.

En ayant recours à un système de gestion distribué, il sera possible de gérer des réseaux de taille importante. Chaque SIGR distribué disposera d'un sous-ensemble d'objets et de tâches relatif au sous-réseau ou à la partie du réseau sur lequel il agit couramment; il aura accès à l'ensemble des objets et informations de tout le "réseau". Suivant l'importance stratégique des télécommunications on pourra très bien voir apparaître des sites de back-up en stand-by prêts à reprendre la gestion du site principal, un peu comme dans l'architecture ONA (IBM).

Détails de fonctionnement d'une passerelle de gestion.

Partant du fait que la passerelle de gestion est un composant majeur, il nous semble intéressant de nous pencher plus longuement sur celui-ci. Pour ce faire, nous avons repris l'approche, axée sur les informations, faite par [PENN94] pour l'intégrateur IDEA[+] dont parle [BHUS94]. Dans ce cadre, nous parlerons d'*unification* plutôt que d'*intégration*.

Cet intégrateur procède à une double unification au sein d'un module d'exécution appelé PROXY par analogie avec SNMP; c'est en fait un des éléments constitutifs d'une passerelle de gestion. Ainsi distingue-t-on l'*unification sémantique* et l'*unification syntactique* (Figure 2-5) [PENN94]. Avant tout, examinons les concepts nécessaires à la définition de ce double mécanisme. Considérons un Monde, c.-à-d. un ensemble d'éléments du réel, potentiels et hétérogènes qui sont appelés des *informations internes* (**ii** - internal information). Ces éléments du réel peuvent être organisés en domaines (par exemple : un LAN et un MAN) qui sont des unités homogènes de l'hétérogénéité du monde (Cfr. Pt 2.4.2). Avec un Modèle, l'utilisateur n'aura qu'une vue externe du Monde. Les éléments de ce Modèle sont des *attributs externes* (**ea** - external attributes); ce sont des représentations abstraites du Monde et des **ii**. Pour accéder aux **ii**, il est nécessaire de localiser et d'extraire leurs champs d'information au moyen d'un *mécanisme d'accès* (**ra** - real access). L'application d'un mécanisme d'accès sur un **ii** permettra l'obtention *d'attribut(s) interne*(s) (**ia** - internal attribute) que nous définirons comme des unités informationnelles d'un domaine, telles que fournies par les constructeurs. Il faut noter que les mécanismes de localisation et d'extraction sont transparents pour l'utilisateur.

Doté de ces concepts, il nous est à présent possible de définir les deux formes d'unifications: l'*unification syntactique* et l'*unification sémantique*. La première fait la transition entre une syntaxe particulière à un domaine et une syntaxe homogène qui constitue la base de l'unification sémantique. C'est donc le processus de passage d'une **ii** à un **ia**. Deux types de relations sont possibles entre ces éléments : ONE-TO-ONE et ONE-TO-MANY (une **ii** se décompose en plusieurs **ia**). La seconde peut, quant à elle, se définir comme le mécanisme permettant le passage d'une vue réelle à une vue formelle d'une information. Elle est donc le mécanisme de transition permettant le passage d'un **ia** à un **ea**. Ici, trois relations sont possibles : ONE-TO-ONE, MANY-TO-ONE (plusieurs **ia** pour former un **ea**) et UNDEFINED (aucun ia ne correspond à cet **ea** dans ce domaine).

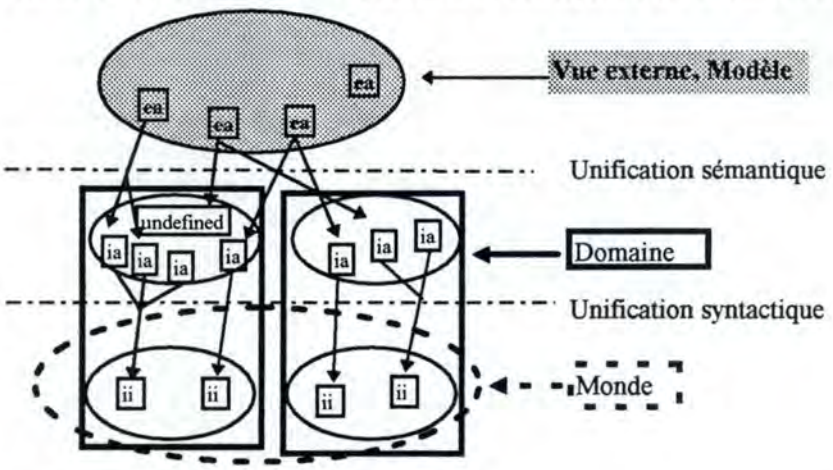
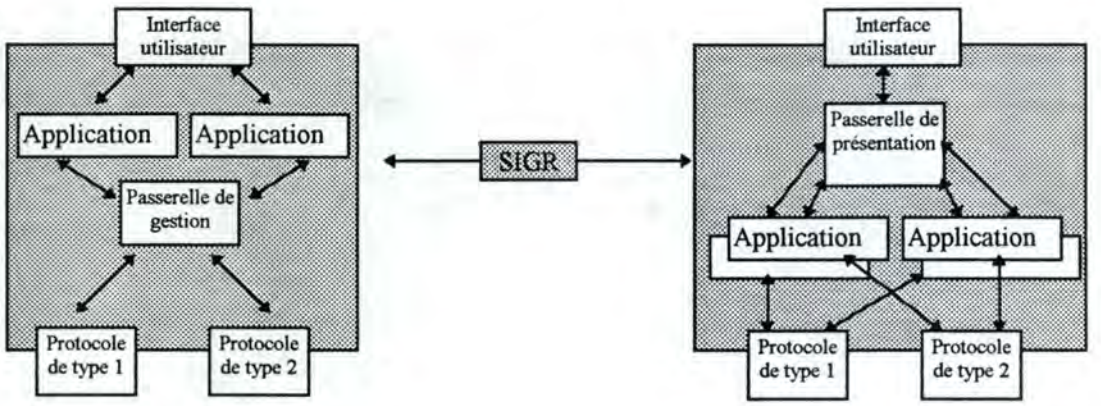


***Figure 2-5** : Méthode d'unification. [BHUS94].*

Il existe un autre moyen pour résoudre les problèmes d'hétérogénéité (Figure 2.6): il est possible d'avoir un gestionnaire multi-protocoles, c.-à-d. implémentant chaque protocole comme si c'était le seul connu et où il existe autant de versions de chaque application que de protocoles. Le SIGR choisit la version de l'application qui correspond à l'élément du réel visé. Dans ce cas, si l'on veut garder une interface unique, il faudra procéder à une unification au-delà de l'application et non en deçà. L'application n'est donc plus générique. Il est évident que cette technique n'est pas du tout économe en ressources.

[+] IDEA = Intelligence, Diagnostic, Expertise, Administration



a) SIGR "classique" *b) SIGR "alternatif"*

***Figure 2.6** : Procédés de résolution de l'hétérogénéité.*

2.3.3. Typologie des approches fonctionnelles d'un SIGR.[BHUS94]

Nous venons de voir que la fonction majeure d'un SIGR était l'uniformisation des commandes et des informations de gestion. [BHUS94] décrit les trois techniques qui permettent de réaliser cet objectif.

Tout d'abord, il y a la technique basée sur un TRADUCTEUR (Translator Based). Dans celle-ci, un système propriétaire est pris comme référence. Il offre des interfaces standardisées où les autres systèmes peuvent venir se connecter. L'intégration se fera au niveau des sous-systèmes qui devront s'adapter au système propriétaire de référence. Cette approche est la plus répandue chez les constructeurs (IBM NetView).

La technique suivante se base sur les STANDARDS (Standards Based). Dans ce cas, tous les éléments du réseau et les systèmes de gestion utilisent un même langage et un même ensemble de fonctions communes. Actuellement il n'y a pas de produit qui suit cette approche puisqu'aucun standard n'a réellement émergé. Il faut remarquer que cette technique fait disparaître la fonction de traduction des SIGR. Il s'agit donc en quelque sorte d'une "méta-intégration" puisque ce sont les systèmes eux-mêmes qui sont uniformisés.

Enfin, l'approche la plus populaire chez les utilisateurs se base sur un INTEGRATEUR (Integrator Based). Dans ce cas, un super-système intègre différents sous-systèmes de gestion (ALCATEL - NM Expert). C'est cette approche que nous avons illustrée dans notre architecture générique; chaque passerelle de gestion représente le procédé intégrateur pour un sous-système de gestion.

Dans une certaine mesure, on peut ramener la technique du traducteur à cette dernière technique. En effet, le traducteur est en quelque sorte la distribution parmi les divers sous-systèmes de l'intégrateur. L'inconvénient majeur sera fort logiquement la multiplication d'éléments semblables; c'est souvent pour atténuer cet effet qu'on a recours à des gestionnaires d'éléments (niveau 2 dans l'architecture fonctionnelle d'un TMN).

Nous devons encore faire remarquer que, si l'on considère que SNMP est un standard (bien que n'émanant pas d'un organisme de standardisation), on peut dire qu'actuellement une certaine standardisation basée sur SNMP est en train de se réaliser. En effet la plupart des composants réseau, LAN et PABX, respectent ce protocole de gestion.

2.3.4. Métaphore du mécanicien .

Afin d'aider à la compréhension de la notion de SIGR, nous allons utiliser une métaphore extérieure au monde des télécommunications. Nous l'avons appelée la métaphore du mécanicien. Avant tout, commençons par établir la correspondance entre les notions de la métaphore et celles du monde des SIGR.

Monde de la métaphore	Monde des SIGR
Voiture	Réseau et ses services
Moteur	Réseau
Manuel de l'utilisateur	Protocole de communication
Pièces moteur	Composants du réseau
Mécanicien spécialisé ou non dans une marque	Opérateur de gestion du réseau

Conducteur	Utilisateur
Boîte à outils	Ensemble des services offerts à l'opérateur par le protocole de gestion
Outil	Service offert par le protocole de gestion
Manuel du mécanicien	Protocole de gestion
Robot mécano	SIGR

Monde de la métaphore	Monde du SIGR
Actuellement, le mécanicien ne sait réparer que la marque de voiture pour laquelle il est spécialisé.	Actuellement, l'opérateur de gestion ne sait travailler que sur un seul type de réseau, celui dont il connaît l'architecture.
Dans sa boîte à outils, il dispose des outils pour réparer tous les véhicules de la marque ainsi que certains autres utilisant les mêmes pièces.	L'ensemble des services offerts à l'opérateur par le protocole de gestion lui permet de réparer ce type de réseau et tout autre pour lequel des services du protocole de gestion sont valables.
Son manuel lui permet de savoir quel(s) outil(s) utiliser pour quelle action. Il lui est impossible de réparer des voitures d'autres marques puisque, outre l'achat de la boîte à outils et du manuel ad-hoc, il doit savoir l'utiliser d'une manière optimale pour chaque véhicule; cela n'est possible qu'avec une certaine expérience.	Les spécifications du protocole lui permettent de connaître quels services sont adaptés aux actions qu'il veut exécuter. Avec ce qu'il possède, il lui est impossible de gérer d'autres réseaux puisque mis à part le fait qu'il doit utiliser un autre protocole, il doit également avoir une certaine connaissance du réseau afin de pouvoir faire des interventions optimales.
Il est clair qu'un mécanicien sans expérience sait se débrouiller en respectant les directives de son manuel mais ne pourra optimiser les réglages. Avec l'introduction du robot, le mécanicien peut réparer toutes les marques de voitures moyennant l'achat de boîtes à outils spécifiques à chaque marque. Elles seront mises à la disposition du robot. Ce dernier aura également en mémoire les manuels du mécanicien qui seront ici électroniques (sa base de connaissances). Suivant le véhicule sur lequel il faut travailler, le robot choisira le manuel et la boîte à outils adéquats.	Un opérateur inexpérimenté sait se débrouiller en se basant sur les spécifications du protocole, mais il est incapable d'optimiser la configuration. Avec l'introduction du SIGR, l'opérateur peut réparer tous les réseaux, pour autant qu'il se procure tous les protocoles (leurs services et fonctionnalités). De plus, il faudra que le SIGR en connaisse les spécifications. Suivant le réseau à gérer, le SIGR aura recours au protocole (au sens large) adéquat.
Si le mécanicien est inexpérimenté, il communiquera avec le robot en utilisant des commandes génériques; le robot est basé sur un intégrateur. Par contre, s'il est expérimenté dans une marque particulière, le robot répondra à ses commandes spécifiques; le robot est alors basé sur un traducteur.	Si l'opérateur est inexpérimenté, il communiquera avec le SIGR au moyen de commandes génériques; le SIGR est dit basé sur un intégrateur. Par contre, s'il est expérimenté sur un réseau particulier, le SIGR répondra à ses commandes spécifiques; le SIGR est alors dit basé sur un traducteur.
Exemple: Si le mécanicien sort de l'école et qu'il doit changer le carburateur d'une Toyota, il dira au robot qu'il doit commencer par dégager les accès au carburateur, puis le débrancher de toutes ses arrivées et sondes. Sur une Toyota cela signifie qu'il faut par exemple enlever le filtre à air, puis débrancher l'arrivée de carburant. La traduction se fait directement à partir des commandes génériques. Si le mécanicien est expérimenté sur une Peugeot, et que le robot le sait, il dira au robot qu'il doit débrancher le débitmètre et l'arrivée d'essence (en supposant que sur la Peugeot le carburateur est directement	**Exemple**: Si l'opérateur sort de l'école et qu'il doit changer les paramètres d'un routeur sous SNMP, il dira au SIGR qu'il doit commencer par lui donner la configuration actuelle, puis modifier les valeurs qu'il lui désigne. Sur un routeur fonctionnant sous SNMP, cela signifie qu'il faut faire des Get jusqu'à la fin du tableau, puis faire un Set contenant toutes les valeurs. La traduction se fait directement à partir des commandes génériques. Si l'opérateur est expérimenté sur un système propriétaire, et que le SIGR le sait, il dira au SIGR qu'il doit effacer la table à modifier (en supposant qu'avec le système

accessible). Le robot ayant en mémoire le manuel de la Peugeot traduira cela en actions génériques: dégager le carburateur après en avoir dégagé les accès. Ces actions seront à leur tour traduites pour exécution de la même façon que ce qui a été fait avec le mécanicien inexpérimenté.

Le robot doit être à même d'apprendre et pourra également aider le mécanicien à accroître sa connaissance au moyen d'un système d'aide en ligne et d'un système expert.

A l'arrivée d'un véhicule en panne, le robot se branchera sur une "prise diagnostic" et analysera les données reçues pour ne transmettre au mécanicien que ce qu'il pense avoir détecté. Celui-ci pourra toutefois consulter l'ensemble des informations unifiées, c.-à-d. mises dans un format standard. Il est donc plus aisé pour quiconque disposant d'un minimum de connaissances, de procéder aux réglages de la voiture. De plus, la connaissance étant disponible pour tous les utilisateurs du robot, personne n'est indispensable et la qualité moyenne du travail est améliorée. Le véhicule est immobilisé moins longtemps et les réglages de celui-ci étant optimaux, les services qu'il rend sont donc améliorés.

propriétaire, on ne puisse modifier une partie d'une table et qu'il faille en réécrire une nouvelle après avoir effacé celle qu'on désire modifier). Le SIGR ayant en mémoire le protocole propriétaire traduira cela en actions génériques: visualiser la table actuelle et modifier les valeurs concernées. Ces actions seront à leur tour traduites pour exécution de la même façon que ce qui a été fait avec l'opérateur inexpérimenté.

Le SIGR doit être à même d'apprendre et pourra également aider l'opérateur à accroître sa connaissance au moyen d'un système d'aide en ligne et d'un système expert.

A l'arrivée d'un évènement, le SIGR fera un diagnostic et analysera les données reçues pour ne transmettre à l'opérateur que ce qu'il pense avoir détecté comme étant un problème. Celui-ci pourra toutefois consulter l'ensemble des informations unifiées, c.-à-d. mises dans un format standard. Il est donc plus aisé pour quiconque disposant d'un minimum de connaissances, de procéder à la configuration du réseau. De plus, la connaissance étant disponible pour tous les utilisateurs du SIGR, personne n'est indispensable et la qualité moyenne du travail est améliorée. Le réseau est indisponible moins longtemps et sa configuration étant optimale, les services qu'il rend sont donc améliorés.

2.4. Modèles.

Nous basant sur [ZNAT94] nous décrirons la gestion de réseau(x) intégrée selon quatre modèles différents. Ceux-ci sont établis en partant du fait que les spécifications d'une application de gestion de réseau(x) doivent déterminer les fonctions qui lui seront associées, son architecture ainsi que celle du réseau, les objets qu'elle va gérer et les moyens de communication associés afin d'arriver à un ensemble cohérent.

2.4.1. Modèle Fonctionnel

Comme nous l'avons vu ci-dessus, l'UIT-T a décrit dans sa recommandation X.700* cinq domaines fonctionnels (gestion FCAPS) que nous nous proposons de voir ici d'une manière un peu plus approfondie. Deux raisons sont à l'origine de la présence de ce modèle dans ce chapitre. Tout d'abord, c'est non seulement l'UIT-T mais aussi d'une manière générale le monde de la gestion de réseau(x) qui considèrent ces concepts comme essentiels. Ensuite, par définition, un intégrateur doit étendre les possibilités de son utilisateur; un SIGR doit donc offrir les mêmes fonctionnalités qu'un gestionnaire classique.

Il y a tout d'abord la **gestion des anomalies**. Dans ce secteur nous pouvons retrouver tout ce qui touche aux problèmes liés au fonctionnement du réseau. Ainsi retrouve-t-on dans ce domaine la réception des notifications ou messages d'erreurs (ou évènements définis comme tels, par exemple : passage de limite par une jauge) et la réaction associée. La recherche d'anomalie(s) par analyse des erreurs et tests de diagnostic, ainsi que la correction des anomalies détectées font également partie de cet espace fonctionnel. Il s'agit certainement de l'activité de gestion la plus opérationnelle.

Ensuite, il y a la **gestion de la configuration**. Elle commence par l'identification des systèmes, continue par leur contrôle et la collecte des informations concernant leur paramétrage (configuration) et se termine par leur éventuelle redéfinition. Diverses fonctions en font partie; citons à titre d'exemples :

- l'établissement des paramètres de fonctionnement normal du système,

* la recommandation X.700 porte sur la gestion de réseau.

- la gestion des noms d'objets gérés,
- l'initialisation et le retrait de ceux-ci,
- la collecte des informations portant sur l'état du système,
- la collecte des demandes de modifications du système et leur exécution.

Il y a encore la **gestion de la sécurité;** c'est probablement l'activité la plus étudiée et avec laquelle beaucoup de gestionnaires sont familiers. On y retrouve la gestion de la sécurité du réseau, de ses composants, des messages y circulant et la gestion des journaux reprenant les "audit-trails" ainsi que les divers évènements relatifs à la sécurité.

Nous profitons ici de l'occasion pour rappeler les divers types de menaces qui peuvent exister [STAL94]. Ils sont au nombre de quatre et peuvent porter sur les entités présentes dans un réseau (données, matériel, logiciel, lignes de communication). La figure 2 -7 représente les types de menaces dans un système:

- INTERRUPTION = un élément est détruit ou devient inutilisable (menace contre la disponibilité).
- INTERCEPTION = un objet non-autorisé accède à une des quatres entités (menace contre le secret) et s'en empare et/ou l'analyse.
- MODIFICATION = un objet non-autorisé non seulement accède, mais en plus modifie une des entités réseau (menace contre l'intégrité).
- DÉGUISEMENT = un objet non-autorisé introduit un élément contrefait dans le système. Il se fait passer pour un autre qui possède d'autres droits (menace contre l'intégrité).

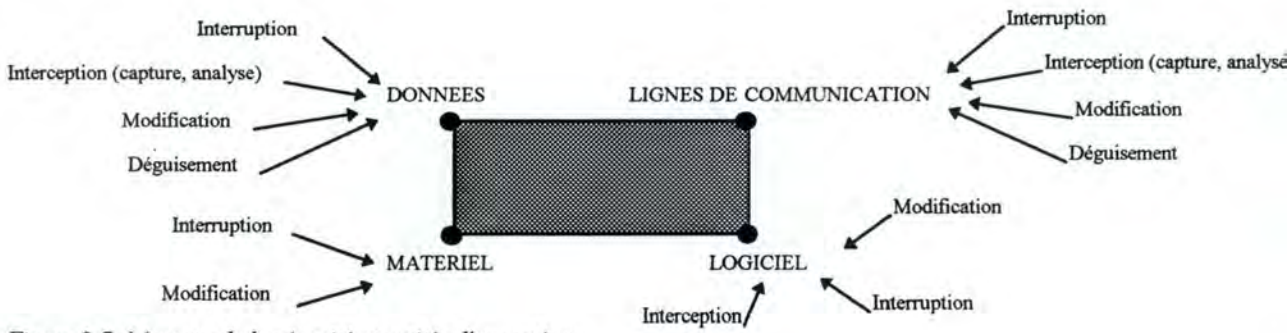


Figure 2-7 :Menaces de la sécurité et entités d'un système.

La **gestion des performances** reprend la mesure et l'analyse du réseau (au point de vue de son efficacité) et de sa qualité de service. A titre d'exemple, citons-en quelques fonctions :

- la collecte de statistiques,
- la gestion des journaux d'état du système,
- la détermination des performances et la simulation,
- la modification des modes de fonctionnement afin d'exécuter des actions de gestion de performances.

Enfin, il y a la **gestion de la comptabilité** dont le but est de gérer les coûts d'utilisation du réseau et des ressources afin d'éventuellement les répartir parmi les utilisateurs. En outre, elle permettra une gestion de la tarification appliquée à l'usage des ressources.

2.4.2. Modèle Architectural. [DISA93]

C'est dans le modèle architectural que se retrouvent les spécificités d'un SIGR. En effet, nous avons vu que dans le modèle précédent, nous reprenions des concepts émanant de la gestion de réseau(x) au sens pur du terme. Dans ce modèle, il n'en est plus question; nous allons d'ailleurs voir où se situe la gestion de réseau(x) par rapport à un gestionnaire de réseau intégré.

Ce modèle nous permet de définir deux concepts de la gestion intégrée: le niveau et le domaine. Au point 2.3.1, lorsque nous nous sommes penchés sur une passerelle de gestion, nous avions alors illustré le concept de domaine sans vraiment le définir. Nous pouvons dire qu'il s'agit d'un ensemble de services et d'équipements qui collaborent afin de fournir aux utilisateurs un ensemble de services de communication. Ainsi, il y a par exemple le domaine WAN (lignes louées et leur modems), le domaine LAN (routeurs, bridges), le domaine des services du système (UNIX) et le domaine de la phonie (PABX). La notion de niveau a également été abordée lorsque, au point 2.0, il a été question d'architecture en couches d'un SIGR. Les niveaux sont ici au nombre de trois (ou plutôt quatre*) alors qu'idéalement, une architecture complète en comporte cinq. Nous avons donc :

- au 1° niveau, les éléments du réseau et les objets gérés; ceux-ci peuvent être contrôlés par des gestionnaires d'éléments gérant un ensemble d'éléments du même type (par exemple : gestionnaire de modems),
- au 2° niveau, les gestionnaires intermédiaires encore appelés gestionnaires de domaine; leur rôle est de contrôler les éléments inférieurs et de coordonner leurs actions intra- et extra-domaine (par exemple : gestionnaire de LAN),
- au 3° niveau, un sous-système dédicacé à la coordination des gestionnaires de domaine, au contrôle du gestionnaire de présentation et à la surveillance du réseau global, c.-à-d. de l'agrégat des quatre domaines (par exemple : centre de contrôle). Il délègue donc les tâches de gestion aux gestionnaires de domaines.

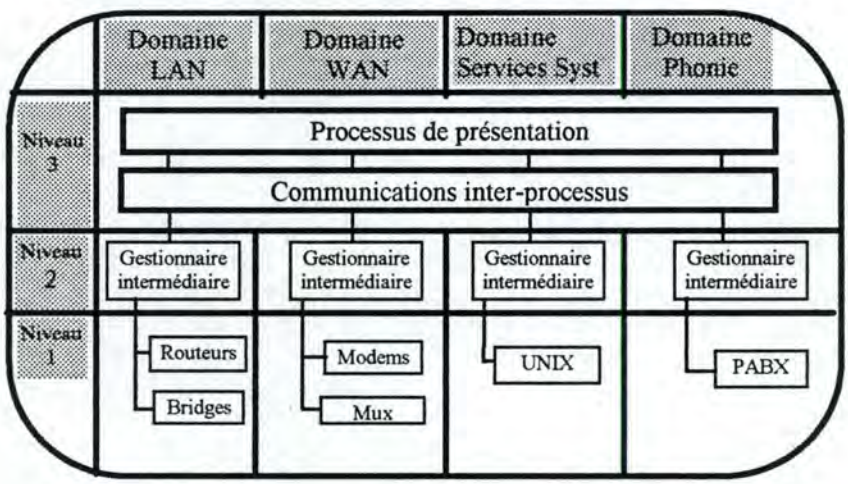


***Figure 2-8** : Modèle architectural d'un gestionnaire intégré.*

Ce modèle est une variante de celui proposé par l'UIT-T. De ce modèle, découlent fort logiquement les deux suivants puisqu'ils portent respectivement sur les informations échangées, les objets gérés et les relations entre niveaux.

2.4.3. Modèle Informationnel

Dans le modèle informationnel, on retrouve tout ce qui a été dit dans le premier chapitre au sujet de la SMI (Structure of Management Information). Ainsi, rappelons que l'on gère des objets en OSI et des attributs (également appelés objets) en TCP/IP. La définition d'un objet OSI encapsule des attributs, des opérations pouvant être effectuées sur ces objets, des comportements de ces objets et enfin des notifications qui peuvent en émaner (Figure 2-9). En TCP/IP les objets sont définis par un nom, une syntaxe, un droit d'accès, un statut et enfin une description. De plus, la SMI définit aussi la MIB; ici on

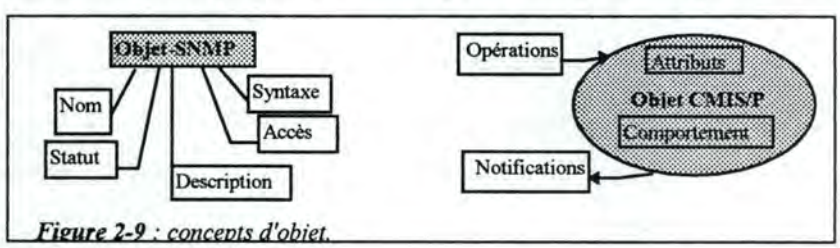


***Figure 2-9** : concepts d'objet.*

* : le niveau 1 regroupe les niveaux 1 & 2 de la recommandation M.3010.

retrouve sous le même nom des choses bien différentes et incompatibles. Notons qu'à l'origine, on voulait pouvoir aisément passer d'une implémentation à l'autre. Certaines études portent d'ailleurs sur les passerelles qui doivent permettre le passage de la MIB-OSI vers la MIB-TCP/IP et vers d'autres standards [ABEC93]. D'autres cherchent à implémenter des agents qui supportent à la fois les requêtes SNMP et les requêtes CMIP [MAZU93]. Ce problème de concordance et de coexistence de standards est une des raisons de l'introduction d'intégrateurs pouvant supporter des agents de types différents.

2.4.4. Modèle Relationnel.

Le modèle relationnel porte sur les relations existant entre les agents et les gestionnaires ainsi que le protocole qu'ils utilisent pour communiquer. Dans ce modèle, CMISE et SNMP se font face. CMISE se compose de CMIS pour ce qui est des services offerts et de CMIP pour les échanges entre entités de gestion (agents et/ou gestionnaire) et est implémenté sur une pile OSI complète. SNMP est quant à lui implémenté sur un service de transport sans connexion (habituellement UDP). Afin de fournir plus de sécurité et d'améliorer les fonctionnalités de SNMP, une deuxième version a été développée et les avantages purement fonctionnels de CMIS/P se sont atténués. Ici aussi, dans le cadre de l'intégration, il faut tenir compte des standards propriétaires qui existaient avant l'avènement de SNMP ou de ceux dont s'est dotée chaque entreprise afin d'accroître sa protection.

2.5. Environnement.

Le but de ce paragraphe est de donner des éléments de réflexion vers d'autres domaines que les télécommunications. A aucun moment nous ne prétendons être exhaustif ou avoir cité tous les auteurs abordant ces différents sujets. Nous pensons que l'avènement des SIGR va de pair avec une simplification des concepts des télécommunications pour l'utilisateur et avec un accroissement des disciplines connexes autrefois plus ou moins ignorées.

2.5.1. Mise en place d'un SIGR.

L'implantation d'un SIGR est un projet dont le degré de complexité et les risques sont élevés. Les nombreuses questions ne peuvent souvent trouver une réponse qu'au moyen d'une collaboration étroite entre les fournisseurs et les utilisateurs, tout comme dans un projet de développement d'un produit. Suivant les critères repris dans la grille de Mc Farlan [LESU95] nous pouvons dire que le risque de ce projet est élevé, voire très élevé, puisque la technologie est souvent nouvelle et l'expérience dans l'entreprise à ce sujet est très limitée. Ceci est logique puisque le but est d'intégrer tout en un seul point, ce qui, par définition, ne peut-être fait qu'une seule fois. Il en sera de même de l'expérience du chef de projet; toutefois, celui-ci sera souvent issu de l'extérieur (de chez le fournisseur) ou aura bénéficié d'une formation lui donnant un maximum de connaissances sur le sujet. L'expérience étant répartie au sein des divers systèmes de gestion existants, il s'agira certainement d'associer un responsable de chacun de ceux-ci à la réalisation du projet. Il nous semble nécessaire d'impliquer au maximum le personnel dans l'élaboration du SIGR. Nous proposerons au point 4.2. une méthode complète d'implantation d'un tel système. En attendant, nous proposons ci-dessous la méthodologie suivie par le personnel de chez BULL pour installer le système ISM.

Un exemple : BULL et ISM (Integrated System Management). [BULL94]

Chez BULL, la mise en place d'un SIGR se fait en trois étapes.

① ETUDE D'ORGANISATION - L'objectif de cette étude est de présenter et d'évaluer les scénarios organisationnels en précisant:

- les critères de regroupement,
- les principes d'organisation,
- les compétences et les profils qui s'en dégagent,
- le positionnement relatif des scénarios par rapport à la stratégie et aux objectifs du projet,
- le nombre de serveurs de gestion,
- d'autres données pertinentes,
- les avantages et inconvénients des scénarios.

Comme dans tout scénario d'organisation, il faut donc répondre à:

Quoi (fonctions d'administration et objets administrés) est administré ***où*** (notion de localisation ou de centralisation) et ***par qui*** (notion d'organisation)?

② DÉFINITION DE LA SOLUTION D'ADMINISTRATION (des réseaux et systèmes) : on retrouve ici plusieurs étapes visant à définir la solution technique envisageable.

- L'étude des besoins et de l'existant recouvre trois domaines essentiels :
  - l'analyse des besoins fonctionnels,
  - l'analyse des fonctions (opérateur, administrateur, technicien) au sein de l'entreprise et l'organisation liées à l'administration,
  - l'analyse détaillée de l'existant au point de vue des systèmes et réseaux.
- L'architecture fonctionnelle déterminera les besoins (fonctions et outils) de chaque utilisateur pour qu'il soit à même de remplir pleinement sa fonction.
- L'architecture technique vise à élaborer des scénarios d'exploitation en confrontant le résultat de la première étape avec l'architecture fonctionnelle et des scénarios théoriques.
- Les scénarios d'exploitation sont évalués sur base de critères pondérés (fiabilité, ergonomie, performance, ...). Un classement des scénarios proposés est établi; à la suite d'une confrontation avec l'ensemble des équipes concernées par le projet, un scénario optimal est choisi. Celui-ci sert de base à l'élaboration de la solution.
- L'intégration de la solution est réalisée moyennant quelques éventuels ajustements selon un plan de migration global élaboré sur base de la solution optimale et des évolutions prévues.

③ IMPLÉMENTATION: dans cette étape, BULL accompagne **le client** dans l'implantation et l'exploitation de la plate-forme ISM en réalisant les prestations de services suivantes :

- Installation,
- Formation des utilisateurs,
- Suivi de site (éventuel).

2.5.2. Organisation de l'Entreprise (OE).

Souvent l'introduction d'un SIGR provoque un regroupement, une centralisation du pilotage des divers réseaux et ressources. Auparavant, ce contrôle était décentralisé ou n'existait pas réellement. Comme nous l'avons déjà dit, les réseaux sont aujourd'hui devenus des ressources critiques pour les entreprises. Certaines en ont d'ailleurs retiré un/des avantage(s) compétitif(s) sur leurs concurrents (par exemple : SABRE et American Airlines). La maîtrise de ces moyens permet certainement de disposer d'un certain pouvoir aux yeux de la direction. Celui-ci se répartit entre le pouvoir formel du responsable du département informatique et le pouvoir informel des spécialistes réseau qui ont acquis une grande expérience. De ce fait, ils sont devenus indispensables ou, en tout cas, leur départ occasionnerait vraisemblablement un problème momentané pour l'entreprise. La décentralisation de la gestion et la complexité de leur tâche placent certainement les experts dans une position privilégiée, par exemple lors de discussions avec le chef de département.

Lors de l'introduction d'un système centralisé, les risques de réticences sont importants puisque le personnel a le sentiment de perdre une partie de son pouvoir [LOBE94]. En effet, la gestion étant simplifiée et l'expérience étant encodée dans des bases de connaissances, le personnel de gestion se pose la question de savoir ce qu'il adviendra de lui, d'autant plus qu'il est possible que cette centralisation s'accompagne d'une certaine réduction d'effectifs.

Si l'on est confronté à un très grande nombre de LAN, le problème sera plus complexe et la réduction de personnel risque d'être plus importante. Pour augmenter le niveau d'accord des divers départements, il serait par exemple possible de montrer au personnel de gestion que grâce à ce système unique, il leur sera désormais possible de pouvoir contrôler tous les réseaux de l'entreprise et que l'expérience ainsi acquise leur sera profitable. Une fois le projet mené à son terme, il nécessite des modifications de la structure organisationnelle qui auront dû faire partie intégrante du projet. Plus les départements y auront été associés, mieux on leur aura expliqué ce qu'il y a à gagner et plus il sera facile de réaliser ces modifications. Diverses approches sont possibles. A titre d'exemple et afin de susciter la réflexion des décideurs, nous proposons le schéma suivant (Figure 2-10) pour choisir le responsable du fonctionnement du système après son implémentation. Dans notre

approche, il sera donné une certaine préférence à un Back-programmer* ou à un chef de projet issu du monde des utilisateurs ou, en tout cas, ayant un contact aisé avec ceux-ci et bien perçu par eux. On le justifie par le fait qu'il risque d'y avoir des remous chez les utilisateurs pour les raisons déjà citées ci-dessus.

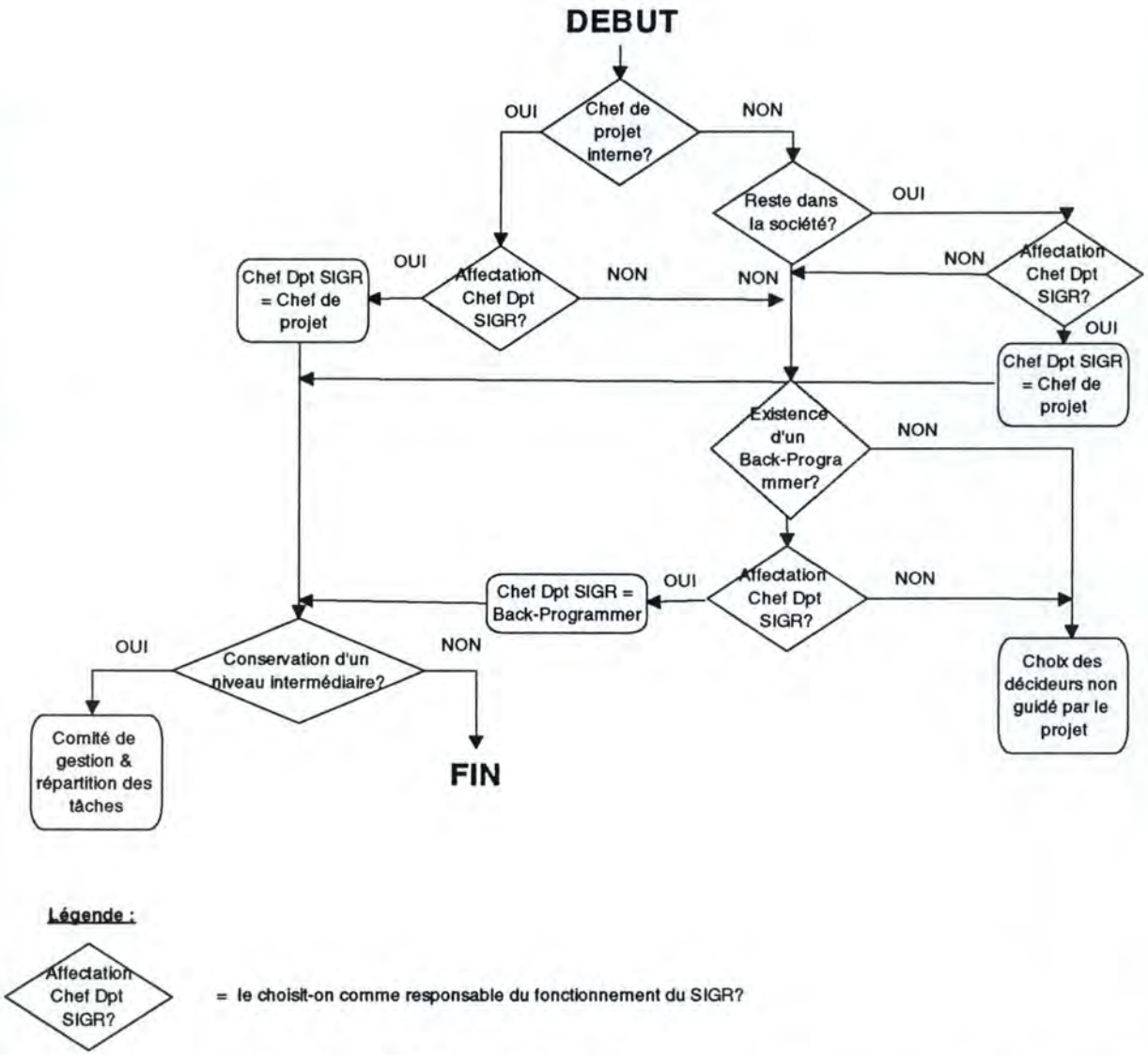


***Figure 2-10** : Ordinogramme d'affectation de la direction du SIGR.*

Si le chef de projet est externe (ou a été engagé pour ce projet), il faut voir si la société désire le conserver en son sein ou dans ses fonctions une fois le projet mis en route. Dans ce cas, il sera désigné comme chef de département SIGR. Il en sera de même pour un chef de projet interne. Si la direction ne désire pas prendre cette option, il faudra voir si un Back-programmer a été désigné. Dans l'affirmative, il est préférable de le désigner comme chef de département. Si la direction refuse cette option, elle devra alors procéder à une désignation qui ne sera pas guidée par le projet. Une fois le chef de département désigné, il faut décider si toute la gestion est reprise par le SIGR ou si ce n'est que la gestion opérationnelle. Dans ce dernier cas, on choisira par exemple de laisser la gestion stratégique aux anciens chefs des départements NMS. Les questions stratégiques et tactiques, voire opérationnelles (a posteriori dans ce cas), seront débattues en comité de gestion regroupant le chef de département SIGR (vue globale des réseaux) et les chefs de départements ex-NMS (vue détaillée à plus long terme). La désignation d'un Back-programmer nous semble être une option à privilégier. Le choix est très délicat et doit certainement tenir compte des risques politiques qu'engendre la mise en place d'un tel système.

* Back-programmer = terme issu d'une théorie de gestion de projet avancée par IBM. Il s'agit en fait d'une personne agissant dans l'ombre du chef de projet pendant tout le développement de celui-ci. C'est lui qui sera chargé de la maintenance du produit au sens large du terme. Le but est qu'il connaisse tous les problèmes de l'élaboration et qu'il ait une expérience du produit avant son utilisation effective.

Le personnel sera en partie repris au niveau du SIGR et en partie laissé au niveau de la gestion stratégique ou affecté à de nouvelles tâches si aucun niveau intermédiaire n'est conservé. Le profil de l'opérateur SIGR sera de préférence un opérateur ayant un esprit d'initiative, un minimum d'expérience et une bonne formation technique de base. L'opérateur de gestion stratégique aura, lui, une bonne formation théorique, une ouverture d'esprit vers les technologies nouvelles et de l'expérience. Pour plus de détails concernant les qualités du personnel occupant les nombreux postes ainsi que son importance, nous renvoyons le lecteur à [TERP92]. Pour synthétiser, on peut dire que le personnel de gestion stratégique fera de la conception alors que l'opérateur SIGR fera de la maintenance.

La vue que nous avons présentée ici fait le lien entre le développement de la solution et son implantation. [TERP92] aborde l'aspect purement implantation en insistant sur le fait qu'il s'agit d'un travail d'équipe et qu'il est difficile de tout gérer à un seul niveau. Il propose quatre groupes de travail qui ne sont pas typés sur les cinq sphères de gestion (FCAPS); ils regroupent en fait les fonctions liées dans le temps et par leur rôle. Ce sont :

- un Network-Operational Center (centre opérationnel de réseau) chargé de la gestion opérationnelle comprenant un service clientèle, un service technique et des opérateurs (≈ fault),
- un Network-Performance Analysis (analyse de performance du réseau) comprenant l'analyste de performance et le coordinateur des BD de gestion (≈ performance).
- un Network Administration (administration de réseau) reprenant la gestion de la sécurité, de la configuration et de la comptabilité (security, configuration, accountability).
- un Network Capacity Design (conception de la capacité du réseau) chargé de la conception du réseau et de la planification de la capacité.

Les deux premiers groupes forment le Network Control Center (centre de contrôle du réseau) tandis que les deux derniers se chargent de la gestion stratégique et tactique.

2.5.3. Interface Homme Machine (IHM).

Il y a un composant du SIGR dont nous n'avons jusqu'à présent pas parlé et qui en est pourtant un élément fonctionnel majeur, c'est l'interface utilisateur. Le but de celle-ci est de permettre à l'opérateur de réaliser son travail le mieux possible, avec précision (minimum d'erreurs) et rapidité, le tout avec un minimum d'efforts et un maximum de satisfaction.

Pour accomplir une tâche, l'homme se construit des modèles mentaux de celle-ci. L'interface a comme but d'adapter la machine à l'homme. C'est ainsi que les interfaces actuelles ont recours aux métaphores qui se veulent être la représentation du fonctionnement de l'application en des termes connus de l'utilisateur. Lorsqu'il s'agit d'une interface spécifique comme celle d'un SIGR, où le type d'utilisateurs potentiels est limité à des personnes ayant un minimum de connaissance télématique, cela permet d'utiliser la métaphore du réseau qui est vraisemblablement proche du modèle mental. L'interface entre l'utilisateur et la machine, c'est l'implémentation de cette métaphore dans un système. Si la métaphore est bien choisie et bien implémentée, elle permet de réduire les distances sémantique* et articulatoire+.[BODA95]

[MEIN91] distingue cinq types de métaphore; tous ne se retrouvent pas nécessairement dans un SIGR mais il est possible d'en retrouver plusieurs selon la tâche accomplie.

LA MÉTAPHORE FONCTIONNELLE: elle rend la machine compatible avec la manière dont l'utilisateur appréhende la tâche (par exemple : arrêter un modem en le visualisant et en poussant sur l'interrupteur).

LA MÉTAPHORE ORGANISATIONNELLE: elle permet l'accès à des informations en visualisant leur emplacement dans l'organisation (par exemple : accéder à partir de la vue globale des réseaux à un appareil déficient situé à un endroit bien précis d'un site au moyen de zooms successifs).

LA MÉTAPHORE INTÉGRANTE: elle est la base même du système puisqu'elle permet d'avoir une vue unique d'applications indépendantes (remarquons que ce n'est pas uniquement au niveau de l'interface que se fait l'intégration).

---

* Distance sémantique = distance ayant trait à la connaissance des objets manipulés et à la signification des commandes et des retours.

\+ Distance articulatoire = distance ayant trait à la désignation des objets et à la "forme", au sens large, des commandes et retours.

LA MÉTAPHORE ASSOCIATIVE: elle permet la navigation par association d'idées (par exemple : l'aide sous forme d'hypertexte).

LA MÉTAPHORE OPÉRATIONNELLE: elle permet d'appréhender l'ensemble des opérations disponibles et les moyens de les utiliser.

Toutes ces métaphores seront d'autant plus efficaces que les concepteurs auront fait une analyse des activités et des utilisateurs afin de connaître et, éventuellement, de trouver un monde qui leur est commun. L'interface sera spécifique puisque les utilisateurs sont typés, et les mécanismes de manipulations seront ceux de standards tels que X Window, de MOTIF ou de Windows.

Un des rôles généralement acceptés des interfaces est l'intégration. (Voir figure 2-11). L'intégration minimum est un gestionnaire de fenêtres commun. Chaque fenêtre émule alors, par exemple, un terminal ou un gestionnaire d'un sous-réseau.

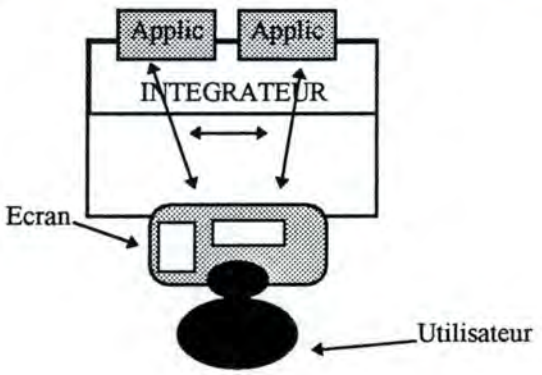


***Figure 2-11** : Intégration entre applications.*

Le type de dialogue utilisé par les SIGR est fréquemment un dialogue objet-action accompagné de la manipulation directe et aussi du langage de commande (à minimiser). On y retrouve également les mécanismes des interfaces graphiques classiques.

Nous retiendrons encore les seize qualités d'une bonne interface citées par [MEIN91]. Celles-ci pourront nous être utiles pour une évaluation ultérieure d'une interface d'un SIGR.

- ☑ *Visibilité* : rendre visible pour l'utilisateur tout ce qui lui est permis et qui est concordant à son profil.
- ☑ *Transparence* : visualiser l'action en cours en des termes proches du monde de la métaphore et non du système. Si le système est distribué, cela inclura le fait qu'il ignore la localisation de la fonction exécutée ou de l'information recherchée
- ☑ *Intuitivité* : permettre à l'utilisateur de deviner le résultat de l'action.
- ☑ *Prévisibilité* : associer toujours le même retour aux mêmes commandes.
- ☑ *Cohérence* (consistance) : rendre prévisible le comportement indépendamment de l'application ou du contexte.
- ☑ *Résilience* : permettre des différences syntaxiques et lexicales par rapport à la commande exacte.
- ☑ *Intégrité* : protéger l'utilisateur et le système (par exemple s'il est distribué) contre toute action irrémédiable aux conséquences désastreuses.
- ☑ *Guidage* : permettre à tout moment à l'utilisateur de savoir où il est, comment se rendre à un endroit et ce qu'il peut faire.
- ☑ *Contrôle* : assurer un feedback clair à chaque action afin de laisser le contrôle à l'utilisateur.
- ☑ *Concision des informations présentées* (ce qui est nécessaire) *et des commandes requises* (raccourci clavier, valeur par défaut ...) : éviter la confusion et les erreurs mécaniques (fautes de frappe).
- ☑ *Cohérence avec le monde réel* : faciliter la construction du modèle mental correct et la compréhension de la tâche à accomplir.
- ☑ *Clarté des écrans et de la présentation* : faciliter la préhension d'une situation.

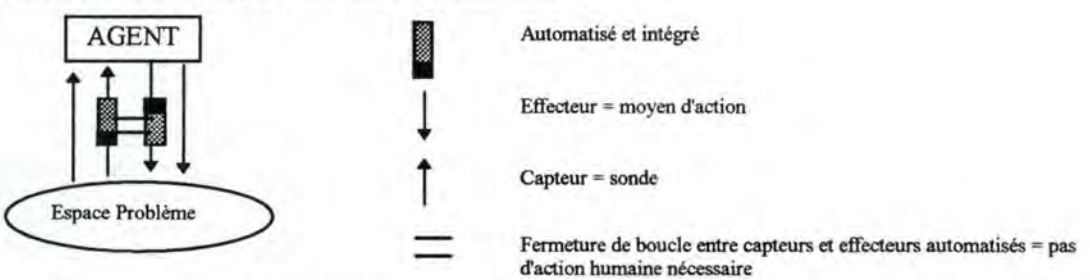
- ☑ *Adaptabilité* : permettre à l'utilisateur d'adapter l'interface à ses désirs, à son image de la situation et ce sans reprogrammation.
- ☑ *Adaptivité* : rendre le système capable de se rendre compatible avec le niveau de son utilisateur.
- ☑ *Automaticité* : permettre d'automatiser les commandes répétitives.
- ☑ *Pression de l'environnement* : ne pas être cause de stress pour l'utilisateur car allant à l'encontre des intérêts de l'environnement (par exemple : causer des retards pour la fourniture de résultats lors d'une situation critique).

Nous terminerons en mentionnant les perspectives d'évolution annoncées par [MEIN91]; certaines sont déjà implémentées dans les SIGR :

- le multimédia permettant, entre autres, l'animation (cela pourrait être utilisé pour rejouer des scénarios précédant une erreur et faciliter ainsi la compréhension d'une cause d'erreur et/ou l'apprentissage par l'opérateur),
- l'hypermédia, par exemple, pour améliorer l'apprentissage,
- la parole et le langage naturel comme langage de commande,
- l'automatisation des tâches répétitives et l'anticipation des actions de l'utilisateur au moyen de macros auto-alimentées,
- la réalité virtuelle, technique déjà employée dans bon nombre de SIGR pour la représentation à l'écran des composants du réseau.

L'interface utilisateur n'est qu'un aspect de l'amélioration de l'interaction avec l'utilisateur, laquelle est étudiée par l'ergonomie cognitive de l'interaction machine-utilisateur. Ainsi, suivant cette discipline, la tâche d'un opérateur sur un SIGR peut être comparée à celle d'un opérateur en salle de contrôle d'une centrale nucléaire. Elle se caractérise par un dynamisme de la tâche et des objets gérés qui ne sont manipulés que par l'intermédiaire d'une machine. La gestion d'erreurs se décompose en surveillance, diagnostic (basé sur le backward reasoning *) et récupération d'incident (sur base de forward reasoning+), tandis que la gestion des performances est en fait un multiaxis control (contrôle multi-directionnel) comprenant à la fois du "tracking" dynamique et statique, c.-à.-d. que l'opérateur tente de maintenir des valeurs à proximité de valeurs cibles respectivement changeantes et stables.[JAVA94-1]

Ce type de tâche est particulièrement étudié en ergonomie cognitive et porte le nom de SUPERVISORY CONTROL [JAVA91-2]. Ces tâches de contrôle ont évolué dans l'histoire et d'un contrôle totalement manuel on est actuellement au stade d'un contrôle semi manuel c-à-d que l'opérateur est assisté par des moyens de perception et d'action intégrés mais sans lien entre ceux-ci.



Remarque : Un agent au sens de la psychologie cognitive sera en principe un opérateur de gestion de réseau(x).

***Figure 2-12** : Contrôle semi-automatisé*

Certains systèmes en sont déjà au stade du semi-automatisé (Figure 2.12) où une réaction automatique se fait sans intervention de l'opérateur et se base sur des capteurs et des effecteurs liés et précis. L'évolution prévue est le *supervisory control* au sens strict. Les boucles d'interaction sont alors généralisées et le rôle de l'opérateur se limite à vérifier le bon déroulement de la gestion. Il procèdera à l'ajustement des valeurs cibles (trimming) voire à leur redéfinition, par exemple, lors d'une modification de l'infrastructure ou de la configuration. Dans les phases les plus critiques qui exigent de la flexibilité et du savoir faire, il en assurera lui-même le pilotage.

* Backward reasoning = partant de la conséquence on veut en retrouver la (les) cause(s).

+ Forward reasoning = partant d'une situation, on recherche les conséquences de l'application successive de diverses opérations. Le but est de trouver une séquence d'opérations dont l'issue correspond au résultat recherché.

Enfin, le stade ultime sera le *supervisory control* de haut niveau où un système expert surveillera les effecteurs et les capteurs. Cela nécessitera des systèmes experts plus évolués et d'autres moyens car l'automation sera alors complète; le rôle de l'opérateur sera limité au contrôle de ce système expert.

Si l'on se réfère à l'étude de Sheridan citée par [JAVA91-2], on peut dire que la gestion intégrée d'un ensemble de réseaux et de ressources rentre bien dans le cadre du *supervisory control*. En effet, si le SIGR ne dispose pas d'un système expert réagissant d'initiative, alors on est face à un *supervisory control* au sens large. Le système ne sert que d'intermédiaire entre l'opérateur et le processus contrôlé et traduit les données complexes afin de pouvoir les afficher d'une manière intégrée. De même, il traduira les commandes en actions de contrôle détaillées et éventuellement multiples. Aucune boucle de contrôle n'est fermée*. Un agent est engagé dans une boucle de contrôle lorsqu'il intègre le feedback de ses actions aux mécanismes qui président à la genèse de ses actions. On retrouve dans tout *supervisory control* un agent avec ses capteurs, ses effecteurs et sa logique de contrôle et un espace problème sur lequel il agit. Si certaines boucles sont fermées, nous sommes face à un *supervisory control* au sens strict.

2.5.4. Système expert - Intelligence Artificielle (SE - IA).

Une des attentes qu'ont les entreprises utilisant un SIGR est l'amélioration du niveau de service. Cela signifie la nécessité d'avoir une réponse rapide, consistante et de qualité. La solution qui vient alors à l'esprit est bien naturellement l'automation de la gestion, mais c'est sans tenir compte de la complexité de la tâche. Celle-ci pose un problème étant donné la flexibilité et l'adaptabilité que doit posséder l'opérateur. Ceci plaide donc en faveur d'un opérateur humain. Il est dès lors souhaitable que certaines tâches soient automatisées; ce sont des tâches routinières et nécessitant une capacité d'attention et/ou de recherche qui dépasse les capacités humaines. Par conséquent, une bonne implémentation de ces systèmes nécessitera d'avoir un niveau d'automatisation approprié (efficace et flexible) des opérations de routines et des traitements de problèmes afin de répondre à la demande et d'augmenter la productivité [MAHA94].

Outre le problème de la capacité humaine, il faut également tenir compte de la rareté et du prix des experts. La formation des personnes est difficile et coûteuse et ne représente donc pas une solution acceptable.

Ainsi, les principales raisons de l'utilisation de systèmes experts et d'intelligence artificielle relevées par [TERP92] sont :

- l'existence d'une demande d'expertise,
- la rareté de l'expertise humaine,
- la difficulté de formation de nouveaux experts,
- des problèmes bien définis devant avoir une solution,
- des erreurs coûteuses.

Architecture d'un système expert.

Un système expert se compose de (Figure 2-13) :

une *base de connaissance* , c.-à-d. un ensemble de règles écrites en PROLOG, en LISP ou encore dans un langage de quatrième génération. L'avenir laisse présager l'éventualité de voir de telles bases de connaissances s'auto-alimenter au moyen de leurs conclusions et du résultat de leur application.

un *espace de stockage temporaire* permettant de conserver les diverses informations à la base du déclenchement et des conclusions tirées (mémoire de travail).

un *système expert au sens strict* (moteur d'inférence) qui sur base des connaissances et des données, va faire une déduction. C'est lui qui va déclencher l'exécution de règles (sélectionnées dans la base de connaissances) et éventuellement demander de plus amples renseignements. [DISA93] nous rappelle les diverses approches possibles pour créer de tels systèmes.

- Le *Rule Based Reasoning* (approche classique) se base sur le paradigme "statut, diagnostic du changement, réponse". Etant donné une situation déjà survenue, un système *Rule Based* exécutera

* Une boucle de contrôle est composée de la logique de contrôle de l'agent et des moyens d'action et de perception. On dira qu'elle est fermée si la perception peut automatiquement entraîner une action sans remonter jusqu'à l'agent pour son déclenchement.

le même raisonnement et arrivera à la même conclusion (il n'apprend pas). Si le problème est en dehors du domaine, il n'y aura pas de solution.

- Le *Case Based Reasoning* se base sur les informations historiques et développe des règles et des stratégies appropriées. C'est une sorte de machine qui apprend, un peu à la manière des experts, pour résoudre un nouveau problème. Ainsi, étant donné un événement, il va rechercher un cas existant ayant les mêmes spécifications d'entrées; s'il en trouve, il fournira la solution et enregistrera son action; sinon il cherchera un cas ayant des similarités avec ces conditions de départ et proposera alors un point de départ au raisonnement. Suite à la décision prise par l'opérateur et sa réaction, le cas sera enregistré dans la base de connaissances.

- Les *réseaux de neurones* qui ne sont toujours qu'au stade de la recherche ont la prétention de modéliser le cerveau humain au moyen de logiciel et de matériel. Ils sont capables de reconnaître et de classer des modèles, ainsi que d'apprendre leur environnement.

Les *informations analysées* peuvent provenir, dans notre cas, des ressources gérées et des utilisateurs, éventuellement via un Help Desk.

La *sortie* de tels processus pourra être une interprétation, un conseil ou une conclusion voire une action; cela dépendra du domaine et du degré d'automatisation du système.

Le système expert pourra être implémenté "on-line", c-à-d analyser les données qui arrivent ou "off-line", c-à-d nécessité que ce soit l'opérateur qui y fasse appel.

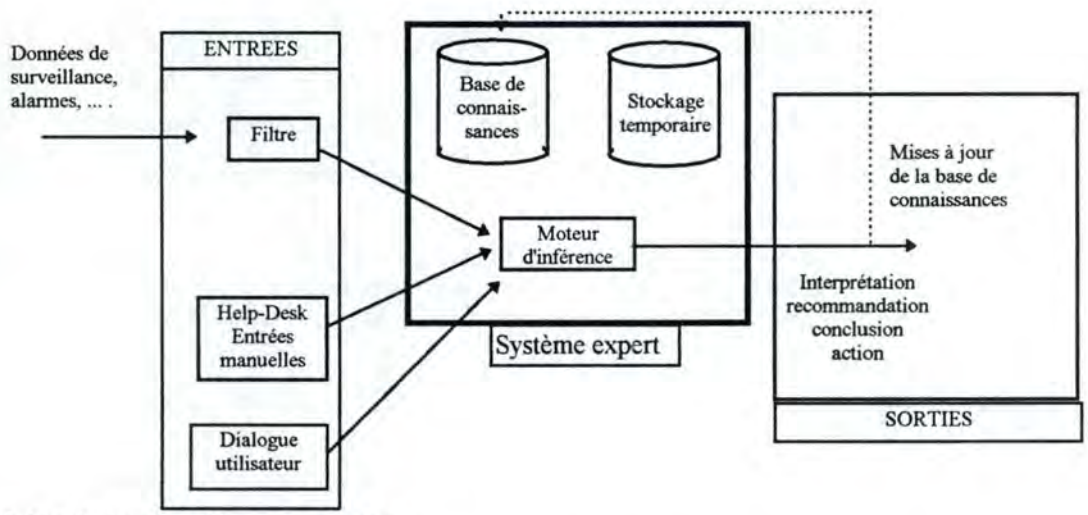


Figure 2-13 : Architecture d'un système expert.

Tous les aspects de la gestion sont concernés par une telle automatisation: la gestion des erreurs, de la configuration ainsi que le planning. On peut également penser à implémenter des systèmes experts pour la gestion de la comptabilité au niveau de la gestion commerciale dans la hiérarchie TMN. De même, il est envisageable d'inclure des "Decision Support Systems" (Systèmes d'Aide à la Décision) incluant des pondérations ainsi que des facteurs et méthodes de décision. Ils seront utiles dans le planning et la gestion de la sécurité. Dans cette dernière sphère, il n'est pas rare qu'il faille prendre des décisions telles qu'une minimisation de l'impact sur le réseau en laissant subsister un problème de sécurité; une autre option consiste au contraire à maximiser les chances de trouver l'intrus, ces recherches ayant alors un certain impact sur les performances du réseau. La solution extrême sera la remise à zéro de toutes les entités concernées.

Les premiers produits apparus concernent fort naturellement la gestion des erreurs puisque c'est aussi la première fonctionnalité offerte par les SIGR. De plus, ce problème est opérationnellement important et nécessite une réponse rapide. Ces produits utilisent en général les messages d'alarme et événements signalant une anomalie. Ils tentent de les corréler avec les règles de leur base de connaissances. L'implémentation de tels produits représente une importante charge de travail, notamment pour la construction de la base de connaissances. Les systèmes experts sont utilisés, entre autres, pour l'analyse, l'identification, la localisation et le diagnostic des fautes. Ils permettent aussi de simuler l'application d'une solution afin d'en permettre l'évaluation.

Les avantages résultant de l'utilisation des systèmes experts cités par [MAHA94] et [TERP92] sont :

- ☑ *Vitesse de détermination des problèmes*: les actions ne requièrent plus d'activités humaines; le système travaille alors à la vitesse de l'ordinateur.
- ☑ *Prise de décision basée sur de nombreuses connaissances ainsi que sur de nombreuses données*: il y a davantage de connaissances dans la base de connaissances que chez un opérateur moyen et une plus grande quantité de données peut être analysée.
- ☑ *Fiabilité*: les tâches répétitives sont exécutées sans interruption.
- ☑ *Diminution du nombre de travailleurs requis*: seuls les incidents n'ayant pas été placés dans la base de connaissances (exceptionnels) doivent être traités par des hommes.
- ☑ *Stabilité*: la réponse aux incidents est consistante et suit la stratégie de gestion.
- ☑ *Dépendance réduite envers le personnel spécialisé*: la plupart des règles ont été implantées dans la base de connaissances.
- ☑ *Flexibilité*: il est relativement aisé de mettre à jour la base de connaissances et les règles.
- ☑ *Interprétation systématique des règles opérationnelles*: il n'y a pas deux manières de comprendre les règles.
- ☑ *Intégration des outils*: les systèmes experts peuvent utiliser divers outils d'extraction de données mais requièrent un type de données pour chaque demande.

Mais ils relèvent aussi certains problèmes tels que:

- ☒ Il faut éviter que la détermination du problème ne soit plus longue que si c'était un opérateur humain.
- ☒ Un grand nombre de règles doit être formulé étant donné le nombre de composants.
- ☒ Les changements dans le réseau impliquent de nombreuses mises-à-jour de la base de connaissances.
- ☒ L'interprétation des messages et leur corrélation peuvent être plus difficiles que prévu étant donné la difficulté de déterminer le début et la fin des symptômes et ceci d'autant plus que l'ordre d'arrivée des messages n'est pas nécessairement équivalent à leur ordre de survenance. Enfin, il peut y avoir un certain délai avant leur arrivée.
- ☒ Les systèmes experts devront avoir une interface avec les nombreux outils nécessaires à la détermination du statut des éléments du réseau.

Nous verrons dans la troisième partie un produit servant spécifiquement à la corrélation d'alarmes.

2.5.5. Base de Données (BD).

Nous avons vu que plusieurs BD existaient au sein d'un même SIGR.

Il y a tout d'abord la MIB de GMO faisant partie intégrante du SIGR. Pour celle-ci une approche orientée objet semble préférable. En effet, cela permet l'encapsulation des méthodes au sein du GMO et la description sous forme d'attributs. C'est d'ailleurs ce que les études sur la compatibilité CMIS/P et SNMP ont proposé en assurant la transformation d'objets (attributs) SNMP en objets CMIS/P. Un modèle global de réseau a été réalisé sur base des *ISO Generic Managed Object Classes* qui ont été quelque peu étendues et de la MIB-II OSI Internet Managed Object Classes qui ont été légèrement modifiées [ABEC93]. Remarquons que rien n'oblige à respecter le standard OSI pour créer la MIB générique.

Selon [TERP92], la BD de gestion de réseau(x) est un élément central de l'implémentation de systèmes de gestion de réseau(x). Certains appellent cela un *repository*. En fait, trois BD sont nécessaires à la gestion de réseau(x), ce sont :

- les données de configuration comprenant les attributs des objets ainsi que leur statut,
- les données de performance avec les indicateurs clés du niveau de service et de l'efficacité,
- les données de télégestion reprenant l'enregistrement détaillé des messages.

A ces BD, certains ajoutent un système d'enregistrement des "trouble tickets" (rapport de problème) permettant une meilleure exploitation ultérieure des problèmes enregistrés. Enfin, il y a encore une autre BD qui est incluse dans le SIGR, c'est la base de connaissances des systèmes experts éventuellement inclus. Suivant les fonctionnalités, il faudra également permettre d'avoir une base de données reprenant les expériences des divers opérateurs et qui doit leur permettre de parfaire leur apprentissage.

A ce jour, l'approche la plus populaire est celle des BD relationnelles, mais les systèmes experts introduisent des BD orientées objet.

Actuellement les BD d'un SIGR constituent des entités indépendantes. Peut-être la solution passera-t-elle par le "directory service" (service de répertoire) sur base de X.500. Dans tous les cas, rien n'est dit quant à l'implémentation physique de toutes ces BD et c'est à l'utilisateur de choisir entre un système centralisé, un système distribué ou encore un système distribué avec un site de back-up centralisé.

2.6. Système Intégré de Gestion de Réseau sur le marché - Distinction entre MoM et plate-forme.

Nous basant sur [TERP94] et sur la tendance du marché, nous pouvons distinguer deux types d'intégrateurs. Ce sont les gestionnaires de gestionnaires (MoM - Manager of Managers), et les plates-formes de gestion. Les MoM se basent sur une organisation hiérarchique (Figure 2-14). On retrouve des gestionnaires gérant les éléments d'un domaine (NMS - Network Management System), d'un réseau, suivant un protocole commun (SNMP, CMIP, Propriétaire utilisant des chaînes ASCII, ...) ou des gestionnaires d'éléments (EMS) d'un même type (par exemple : modems). Au-dessus de ceux-ci, il existe un gestionnaire (SIGR) qui communique avec ces gestionnaires d'éléments ou de domaines suivant un protocole ne devant pas nécessairement être commun aux NMS. Ce MoM est chargé de la coordination des gestionnaires et de toutes les fonctions dévolues à un intégrateur. Dans le cas des plates-formes (Figure 2-15), les gestionnaires d'éléments et de réseaux sont des modules logiciels, des applications, qui interagissent avec les composants réseau et qui peuvent éventuellement être distribués (plate-forme distribuée). La couche supérieure de la plate-forme est composée des services de gestion et représente la plate-forme à proprement parler. Entre ces deux composants, nous trouvons diverses API (Application Programing Interface). Celles-ci sont proposées par les constructeurs de la plate-forme et fournissent un point d'accès pour permettre l'intégration des applications de gestion développées par les constructeurs de composants ou par les utilisateurs de la plate-forme. Une des API les plus courantes est XMP API*. Souvent, les plates-formes se composent également d'une interface vers une base de données (par exemple : SQL). Elles reposent sur un OS tel Unix ou DOS. Bien entendu, il ne faut pas oublier les interfaces de communications reprenant les différents protocoles utilisés entre gestionnaire et agent. D'une manière synthétique, on peut comparer une plate-forme à un langage de programmation dont certaines fonctions sont implémentées et dont d'autres doivent l'être par l'utilisateur. Elle ne nécessite pas l'existence préalable de systèmes de gestion. Elle permet l'adjonction de divers modules pour assurer la gestion des environnements existants. Un MoM repose quant à lui sur des systèmes de gestion préexistants et se contente d'intégrer les fonctions offertes par les NMS dans un environnement cohérent et unique. Pour simplifier, on peut dire qu'un MoM au sens strict est incapable de fournir des fonctions supplémentaires à celles offertes par le NMS d'un réseau déterminé. Les seules nouvelles fonctionnalités seront issues de l'intégration des divers environnements (facilité de corrélation). Terminons en ajoutant qu'à l'heure actuelle certains MoM autorisent l'adjonction de modules logiciels qui permettent ainsi d'offrir de nouvelles fonctions ou encore de directement gérer des environnements qui ne sont pas munis de NMS, ce qui les rapproche des plate-formes qui pour leur part prennent en considération les systèmes de gestion préexistants.

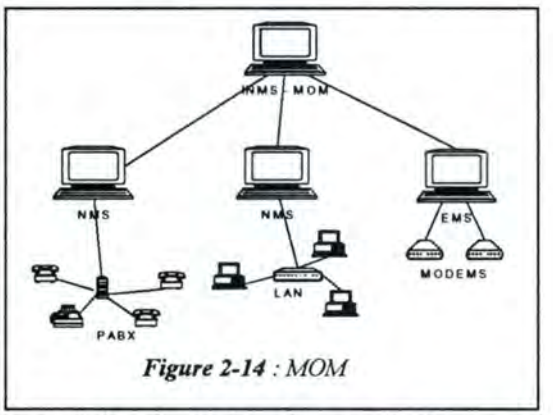


Figure 2-14 : MOM

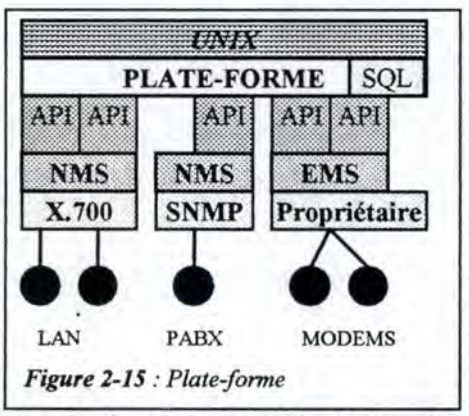


Figure 2-15 : Plate-forme

* XMP API = X/Open Management Protocol API a été choisie par l'OSF DME (voir Ann D) comme API de gestion pour CMIS/P et SNMP et permet l'utilisation des services de SNMP et de CMIP par des applications de gestion de système.

D'une manière plus pragmatique on peut dire qu'à l'heure actuelle, les plates-formes sont plus courantes que les MoM. Il faut savoir que la plupart des constructeurs ont tendance à pousser de tels produits puisqu'ils les libèrent de la charge que représente le développement de leur propre gestionnaire; ils peuvent développer les applications de gestion de leurs produits en se basant sur un gestionnaire courant. Il leur suffit d'écrire une ou deux applications utilisant les API de la plate-forme pour assurer la gestion de leur produit. De leur côté, les utilisateurs désireraient voir se développer des MoM car il n'est pas rare qu'ils aient déjà acquis l'un ou l'autre gestionnaire qu'ils désirent pouvoir conserver.

Dans la seconde partie nous aurons l'occasion de rencontrer divers produits, certains sont des MoM et d'autres des plates-formes. Nous renvoyons le lecteur à cette partie pour plus de détails pratiques.

PARTIE 2 : Approche pratique.

Chapitre 3 : Etude de systèmes existants.

3.0. Introduction

Dans ce chapitre, nous visons un double objectif. Tout d'abord nous dresserons un tableau synoptique reprenant les produits les plus courants en Belgique. Ce tableau sera complété par un ensemble de fiches descriptives de chacun de ces systèmes. Ensuite nous nous pencherons davantage sur deux produits. Le premier est une plate-forme tandis que le second est plutôt un Manager-of-Manager (MoM).
Le tableau synoptique que nous proposons n'est pas le fruit d'une expérience pratique. L'idéal eut été de tester chacun de ces produits sur un ensemble de réseaux au sein d'une entreprise. L'étude approfondie d'un de ces systèmes et de son implémentation peut, à elle seule, faire l'objet d'un mémoire.
Le tableau que nous dresserons sera avant tout un moyen permettant d'avoir un regard homogène sur la documentation que nous avons pu rassembler auprès des constructeurs. Notre approche de ce sujet est intéressante puisqu'elle est inédite. Il faut savoir que les gestionnaires intégrés sont en pleine expansion et qu'il est aujourd'hui difficile de savoir rapidement à quoi correspondent ces divers produits. Nous nous sommes donc essentiellement basé sur un questionnaire soumis aux constructeurs[+] (Cfr. Annexe C) et sur la documentation commerciale qu'ils nous ont fait parvenir.

3.1. Etat du marché

Comme nous l'avons déjà dit dans l'introduction de ce chapitre, ce point offre au lecteur une vue homogène sur la documentation pouvant émaner des divers constructeurs. Ce tableau ne prétend pas être exhaustif, mais il présente la plupart des produits existants en Belgique. L'intérêt est d'ainsi présenter des systèmes pour lesquels il existe une infrastructure d'assistance et d'installation dans notre pays.

Outre le tableau synoptique, nous dresserons une carte de visite de chacun de ces produits en dix points. Ceux-ci nous ont d'ailleurs servi de base pour l'élaboration de notre questionnaire (Cfr. Annexe C). Ce dernier peut également être utile pour analyser toute documentation "commerciale" d'un tel système. L'élaboration du questionnaire est issue d'une synthèse de [WIER94], d'un document interne de la BBL[*] , d'un document émanant de l'université du Michigan [FCEM94] ainsi que de notre analyse personnelle.

3.1.1. Remarque préliminaire.

La plupart des produits proposés par les grands constructeurs se composent en fait d'une plate-forme et d'un ensemble de modules. La plate-forme fournit les fonctions de base utilisées par les modules qui sont en fait des réponses aux demandes fréquentes des utilisateurs. Souvent, outre ces modules parfois très nombreux (par exemple : la gamme HP OpenView), il est possible de faire remplir n'importe quelle fonction au système en ajoutant des modules développés spécifiquement pour nos besoins. La différence majeure réside dans la façon dont on peut développer ces modules et fonctions (langages, API disponibles, compatibilité avec des produits de tiers,...).

3.1.2. Description des éléments de l'analyse.

Ci-dessous nous présentons les neuf points que nous avons retenus et tentons d'expliquer en quelques mots les notions et concepts qu'ils regroupent.

- ARCHITECTURE. Nous parlerons ici de l'O.S. sur lequel tourne le SIGR et de l'organisation (distribuée, client/serveur). Si possible, nous donnerons une description de l'organisation logique des différents modules fonctionnels.

- DOMAINE(S). C'est dans ce point que nous verrons à quoi peut servir le système étudié. Ainsi, nous verrons le(s) réseau(x) qu'il permet de gérer et les possibilités de gestion de Base de Données (BD) et des systèmes. En effet, les SIGR (Système Intégré de Gestion de Réseaux) sont couramment appelés plates-formes de gestion d'entreprise et servent à la gestion de toutes les ressources informatiques d'une entreprise. Bien que nous nous limitions à la gestion des réseaux, il nous semble utile de ne pas ignorer ces possibilités puisque, comme nous le verrons au point suivant, il est nécessaire de tenir compte des futurs besoins potentiels.

[+] Sur 6 questionnaires envoyés, seuls 2 ont aboutis et 1 de ceux-ci ne répondait pas aux questions.

[*] BBL = Banque Bruxelles Lambert.

- EXTENSIBILITÉ. Ce point est d'une importance capitale. En effet, nous savons tous que l'informatique se caractérise par d'innombrables évolutions; or une plate-forme de gestion est un investissement important qui est fait pour le moyen terme (5-10 ans). Il s'agit donc de pouvoir faire évoluer le système afin qu'il continue à pleinement remplir ses fonctions. Cette extensibilité devra de préférence se faire aisément, faute de quoi les coûts de fonctionnement risquent de grimper en flèche.

- INTERFACE (DE GESTION DE) RÉSEAU. Nous avons vu dans la première partie les deux grandes familles de protocoles de gestion. C'est dans ce point que l'on va retrouver celui ou ceux utilisable(s) par le SIGR. Nous verrons également les interfaces de communication qui sont offertes sur le système étudié.

- COMPATIBILITÉ AVEC D'AUTRES STANDARDS. Le monde des télécommunications est fait d'innombrables standards. Tantôt ils émanent d'organismes de standardisation plus ou moins officiels (UIT-T), tantôt ils sont l'oeuvre d'associations de constructeurs et/ou d'utilisateurs (OMG[+]). Dans ce point nous verrons ceux qui sont pris en considération par le système.

- GESTION DE DONNÉES. Outre la possibilité de gérer des BD avec le système, il est logique d'envisager la compatibilité des données de gestion avec l'un ou l'autre SGBD (Système de Gestion de Base de Données).

- INTERFACE UTILISATEUR. C'est une autre caractéristique majeure des SIGR. Nous y accorderons beaucoup d'attention puisque c'est avant tout par ce moyen que l'opérateur a une vue intégrée et unifiée de l'ensemble des systèmes. Nous verrons les vues offertes par le SIGR et les possibilités de personnalisation.

- FONCTIONNALITÉS. Ce point représente, on s'en doute, la pièce maîtresse de cette analyse. Toutefois, il faut relativiser les possibilités des SIGR, car rappelons-le, nous nous basons sur la documentation des constructeurs et/ou vendeurs et non sur des tests pratiques. Nous mettrons l'accent sur la gestion des erreurs et de la configuration étant donné qu'il s'agit là des fonctions les plus opérationnelles.

  La gestion de la sécurité sera peu analysée car elle peut être ramenée, au point de vue opérationnel, à la gestion d'erreur. En effet, toute menace ou violation qui est détectée est un évènement qui fait l'objet d'un message et est géré comme une erreur classique. Ce qui précède est valable pour autant qu'une application de surveillance puisse venir s'intégrer dans l'ensemble des applications de gestion.

  Un autre critère concerne la gestion des performances et leur analyse a posteriori. Nous nous pencherons sur les formes et moyens de présentation des données.

- DIVERS. Toutes les caractéristiques du système qui ne sont pas reprises dans un des points précédent seront commentées dans ce point-ci.

REMARQUE : c'est volontairement que nous n'avons pas pris les coûts en considération. Cela risquait d'entraîner des conclusions hâtives concernant certains produits qui à première vue, auraient été trop bon marché ou trop coûteux. L'accommodation d'un SIGR à une situation étant toujours nécessaire, il n'existe pas de solution standard.

[+] OMG = Object Management Group Inc. . Il s'agit en fait d'une organisation sans but lucratif composée de vendeurs, d'utilisateurs et de développeurs qui cherche à promouvoir la technologie Orientée-Objet. Une de ses principales réalisations est CORBA (Common Object Request Broker Architecture).

TABLEAU SYNOPTIQUE DES PRINCIPAUX SIGR

Vendeur	Nom du Système											
Architecture	Domaines	Protocoles de Gestion	Protocoles de Comm	Autres Standards	SGBD	IHM : Vues	IHM : Messages	Gestion des Fautes	Gestion de la Configuration	Gestion des Performances	Gestion de la Sécurité	Gestion de la Comptabilité
Alcatel	**NM Expert**											
(D) - C - SE	E	S - C - A - (T)	X - A - TI - SNA	(OV) - SNM	SQL =I, O, S (T)	C - LPO - A	IT-A	☺	☺	☺	☺	☺
Boole & Babbage	**Command Post**	**3.1**	**&**	**Auto Command**								
D - CS	E	O	?	OV - D	SQL = S	C - LPO - A	IT - A	☺	O	☺	☺	O
Bull	**ISM**	**3.0**										
D - CS	E	(T)	TI-SNA-X-E-F-N-+	D - A+	SQL = O	C - LPO -A	ITS - A	☺	☺	☺	☺	O
Cabletron	**Spectrum**	**3.0**										
D - CS - SE	E	S - (T)	SNA - (T)	OV - SNM - NV - NMS	SQL = I ASCII	C - LPO - A	ITS - A	☺	☺	☺	O	O
Digital	**Polycenter**	**products**										
D - C	L - W - V	S - S2 - C	X - ATM - E - TR - F - I - TI - A	OV - D - O1 - CO	SQL	C - LPO - A	IS - A	☺	☺	☺	☺	☺
Hewlett Packart	**Openview**	**3.0**										
D - C	L-W-S-(T)	S - C	TI	OV	SQL = I ASCII	C - LPO - A	I - A	☺	☺	O	O	O
IBM	**NetView**	**6000**										
D	L-W-(T)	S - C	TI - OSI - +	D - NV	SQL =I ASCII	C - LPO - A	IT - A	☺	☺	☺	O	O

TABLEAU SYNOPTIQUE DES PRINCIPAUX SIGR (Suite)

Vendeur	Nom du Système											
Architecture	Domaines	Protocoles de Gestion	Protocoles de Comm	Autres Standards	SGBD	IHM : Vues	IHM : Messages	Gestion des Fautes	Gestion de la Configuration	Gestion des Performances	Gestion de la Sécurité	Gestion de la Comptabilité
SunConnect	**Net Manager**											
D - C	L - M - W	S - S2 - (C)	X - SNA - A - E - F	O1	(SQL) ASCII	C - LPO - A	IT - A	☺	😐	☺	O	O
Telindus	**TOM**											
D - C	T - (E)	S - C - +	X - N - +	D	SQL ASCII	C - LPO - A	IIS - A	☺	☺	☺	O	O

Légende : Généralités: + = Voir descriptif
( ) = possible mais pas en standard

Architecture: D = Distribué
C = client/serveur
SE = système expert

Domaines: L = LAN
M = MAN
W = WAN
T = Tous les réseaux.
E = Entreprise (T & système)

Protocoles de gestion: S = SNMP
S2 = SNMPv2
C = CMIS/P
T = Tous (y compris propriétaire)

Protocoles de communication: TI = TCP/IP
N = Novell NETWARE
A = ATM
E = Ethernet
TR = Token Ring
F = FDDI
I = ISDN
T = Tous

Autres standards: OV = OpenView
D = OSF/DME
O1 = OMNIPoint 1
SNM = SunNet Manager
NV = Net View
NMS = Novell NMS
A = Applicatif (Gestionnaire de composant)

SGBD: ASCII = ASCII
SQL = SQL I = Ingres
O = Oracle
S = Sybase
T = Tous les SGBD

IHM Vues: C = Carte
L = Logique
P = Physique
O = organisationnelle
A = Adaptable

IHM Messages: I = Icône modifiée
T = Textuel
S = Sonore
A = Adaptable

Gestion (toutes): ☺ = Bon
😐 = Moyen
O = Pas prévu

3.1.1. ALCATEL NM-Expert.

ARCHITECTURE.

La base de connaissances et le système expert qui l'exploite sont le coeur de NM-Expert.

NM Expert est orienté client/serveur dans le sens où le noyau doit être localisé à un seul endroit, sur une seule et même machine. Ceci a comme avantage d'avoir une seule représentation des connaissances de gestion localisée dans la base de connaissances. Ainsi, toute application construite avec NM-Expert aura accès aux données de la base de connaissances. Cela en facilite bien évidemment la cohérence. Les autres bases de données (BD) utilisées pourront quant à elle être distribuées dans la mesure où leur système de gestion de base de données (SGBD) le permet. De même tous les modules d'interfaçage avec l'extérieur peuvent également être distribués sur diverses machines. NM-Expert peut communiquer avec toute sorte de BD, indépendamment du SGBD utilisé, pour autant qu'une interface de base de données ait été écrite et adjointe au système. Le principe est le même en ce qui concerne les protocoles de communication et de gestion pouvant être utilisés. Certains standards font parties du "package" de base et sont disponibles.

Il est possible d'organiser les divers systèmes NM-Expert se partageant la gestion sous une forme hiérarchique ou purement distribuée (Peer-to-peer). De plus NM-Expert offre la possibilité de gérer les ressources à partir de sites distants connectés au noyau au moyen d'un WAN (TCP/IP).

Pour plus de détails nous renvoyons le lecteur au point 3.3.

DOMAINE(S).

L'utilisation de NM-Expert n'est pas réduite à la gestion des éléments de télécommunication, mais peut également servir pour gérer les applications et services distribués sur le(s) réseau(x) ou encore tout matériel gérable à distance (système de climatisation, d'ascenceur, ...). Ce système est indépendant de tout domaine et peut, si cela est nécessaire, corréler des informations concernant des domaines différents. Ainsi, il pourra simultanément gérer une application offrant un service EDI et accéder aux informations de gestion de domaine sur lequel celle-ci repose comme, par exemple, les services de messagerie de type X.400 ou SMTP.

NM-Expert permet l'accès et la gestion des bases de données (de type SQL étendu) soit par l'intermédiaire de modules d'interfaçage avec le SGBD qui offrent la possibilité d'interagir avec la BD au moyen de requêtes logiques, soit plus traditionnellement au moyen d'émulateurs.

Rien n'est dit à propos de la gestion de systèmes.

EXTENSIBILITÉ.

NM-Expert n'est certes pas aussi répandu que HP OpenView et ne dispose pas de ce fait d'autant de développement; son architecture permet de l'étendre relativement aisément. En effet, il est possible de disposer d'une interface avec des applications externes dont l'appel serait fait à partir de n'importe quel module de NM-Expert. Cet appel consistera en un échange de messages (via TCP/IP) ou plus classiquement au moyen d'un appel direct à un programme UNIX (éventuellement dans une règle de la base de connaissances). Les applications seront écrites en C.

L'ajout de règles dans la base de connaissances se fait très aisément soit en ligne soit au moyen d'une interface appelée KAMS (Knowledge Acquisition and Maintenance System).

INTERFACE (DE GESTION DE) RÉSEAU.

Pour communiquer avec les objets et ressources, NM-Expert utilise une interface réseau (Network Interface - NI) qui se compose de deux parties. La première transforme les bytes en objets de gestion et a pour fonction de cacher au système les détails du protocole. La seconde est composée de plusieurs modules spécialisés écrits en C, chacun traitant un type de réseau donné. Ils s'occupent des tâches de communication de données spécifiques à un réseau ou à un protocole. En cas d'ajout d'un nouveau protocole, il suffit d'écrire un de ces modules. Il assurera la transition entre la première partie générique et commune à tous les protocoles, et le nouveau protocole.

Actuellement les modules suivants ont été développés :

- Ligne série asynchrone,
- X.25,
- TCP/IP,
- SNMP,

- SNA,

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

La compatibilité avec les autres standards et les autres plates-formes est assurée par l'entremise de ces mêmes modules. Ainsi actuellement, la compatibilité avec SunNet Manager et OSF/DME (HP OpenView) est assurée ou en passe de l'être.

GESTION DE DONNÉES.

NM-Expert ne maintient aucune base de données et est de ce fait totalement indépendant de tout SGBD; toutefois, l'utilisation d'un SGBD est nécessaire pour diverses applications ou fonctionnalités (par exemple : reporting). Il est donc nécessaire de pouvoir accéder à des BD qui lui sont externes. Ces accès sont possibles par l'intermédiaire de modules d'interfaçage également écrits en C. De telles interfaces existent actuellement pour les SGBD INFORMIX, ORACLE et SYBASE.

INTERFACE UTILISATEUR.

L'interface repose sur un système de fenêtres basées sur X Window et OSF/Motif; elle est donc de type graphique. Elle permet l'affichage des problèmes et des scénarios déclenchés pour leur traitement. Elle permet aussi d'interagir avec les divers réseaux gérés ainsi que d'en modifier la configuration. De même, l'interface facilite l'interaction avec le SIGR. Enfin, elle autorise l'affichage des réseaux et de leurs états sous une forme graphique.

La personnalisation de l'interface est possible. Tout d'abord, il faut savoir qu'elle est automatiquement adaptée à l'information contenue dans la base de connaissances. Cela permet d'éviter, dans bon nombre de cas, l'usage d'un outil GUI (Graphical User Interface - Interface Utilisateur Graphique). Les icônes peuvent être personnalisées en utilisant un éditeur graphique. Enfin, un très grand nombre de paramètres seront personnalisés au moyen des fichiers de configuration. Notons qu'il est toujours possible d'ajouter des fenêtres et commandes à celles qui sont fournies par défaut, mais cela requiert une connaissance approfondie de X Window et de OSF/Motif.

FONCTIONNALITÉS.

NM-Expert se caractérise par deux grandes fonctionnalités : le traitement d'évènements et la gestion de la configuration. Ainsi, on peut dire qu'elle est naturellement orientée vers la gestion de fautes qui se base sur les évènements et vers la gestion de la configuration grâce à l'interface. Ses moyens d'interaction avec des bases de données, son fonctionnement en temps réel et la possibilité d'ajouter de nouvelles applications rendent possible la gestion des performances et de la comptabilité. Enfin, à condition d'avoir un système de sécurité qui génère des messages à destination du SIGR, il sera possible de faire de la gestion de la sécurité.

A priori rien n'est prévu pour la gestion des performances, de la sécurité et de la comptabilité.

FORMATION DU PERSONNEL.

D'après nos informations, on compte habituellement une semaine pour former les développeurs à l'utilisation des mécanismes de la base de connaissances et des interfaces standard fournies actuellement.

DIVERS.

Les commandes qui peuvent être émises sont des commandes logiques, c.-à-d. qu'elles regroupent plusieurs commandes physiques.

AVANTAGE(S) ET INCONVÉNIENT(S).

☺ Encodage des connaissances au moyen d'un langage de haut niveau.

☺ Existence d'un système expert.

☺ Grande ouverture tant au niveau des applications que des bases de données et des protocoles.

☺ Capacité potentielle de gestion de tout équipement (y compris en dehors du monde des télécommunications, par exemple : ascenceurs, climatisation).

☺ Modélisation des objets et de leurs relations sous une forme proche du modèle ERA (Entities Relationship Attributes).

☺ Nombreuses personnalisations.

⊗ Peu d'utilisateurs (relativement), d'où des interfaces peu nombreuses.

3.1.2. BOOLE & BABBAGE - COMMAND/Post.

ARCHITECTURE.

Command/Post repose sur une architecture distribuée client/serveur. La communication entre les clients et les serveurs est assurée au moyen d'un LAN Ethernet (802.3) à 10 Mb/s. L'architecture physique est la suivante. Les serveurs et les clients sont respectivement des systèmes sous UNIX et des stations de travail également basées sur UNIX. Outre ces éléments, on retrouve également des serveurs de communications dont le rôle est de transformer les informations de type ASCII arrivant sur une porte de type RS-232 au format TCP/IP. On y ajoutera dans certains cas des convertisseurs de protocole afin de permettre la communication avec des systèmes pour lesquels il n'y a pas d'émulateur Command/Post existant. Enfin, afin d'assurer la communication avec des systèmes de gestion d'éléments tournant sur des PCs, on retrouvera des PCs et des stations de travail équipés avec la technologie Omni-Ware. L'architecture logique se compose de : l'Operating System UNIX (Solaris ver 4.1.3.), de l'interface utilisateur graphique basée sur le langage Smalltalk-80 et d'un gestionnaire de fenêtres du type OSF/Motif ou OpenLook, d'un système de gestion de base de données relationnelle (SYBASE de Sybase Corporation) sur le serveur Command/Post et d'un manipulateur d'évènements qui est le cerveau du système. Ce dernier reçoit les demandes des diverses fonctions et oriente les informations qu'il reçoit vers les "utilisateurs" qui en ont fait la demande.

En utilisant des "bridges" Ethernet, il est possible d'accéder aux serveurs à partir de sites distants.

DOMAINE(S).

Command/Post permet de gérer l'ensemble des ressources informatiques d'une entreprise. Pour assurer cette gestion, Command/Post se connecte aux divers gestionnaires d'éléments et d'applications en utilisant la technologie RS-232, les câbles coaxiaux, Ethernet, Token Ring, ...

EXTENSIBILITÉ.

Rien n'est dit à ce sujet dans la documentation. Toutefois, d'après les informations que nous avons pu recueillir, il semble que ce produit soit en plein développement. Ainsi, un premier module pour automatiser les réactions est-il déjà venu s'ajouter (Auto-Command). Toutes les fonctions sont exécutées au moyen d'émulations; il est ainsi à priori possible d'ajouter une application qui se servira, par exemple, des évènements enregistrés afin de remplir une fonction. Les scénarios écrits au moyen d'Auto-Command permettront d'offrir une intégration quant à la présentation et de masquer les multiples émulateurs utilisés. La complexité du langage REXX*, relativement peu connu des programmeurs, rend cette construction difficile. Il semble qu'une interface soit en cours d'élaboration afin de permettre la construction de scénarios REXX sans connaissance préalable.

INTERFACE (DE GESTION DE) RÉSEAU.

Etant donné que Command/Post est un manager de manager (MoM), il n'existe aucun moyen d'action directe sur des éléments du réseau. Avec la version 3.1, sont apparues les unités de connexions; elles détaillent les connexions physiques et logiques à divers ordinateurs, gestionnaires d'éléments, plates-formes SNMP et systèmes de communications vocaux. Nous ne citerons que les plus connues : les unités de connexion avec les routeurs Cisco, Novell NetWare 3.11, SunNet Manager, ... On y retrouvera les formats détaillés des messages émis et des alertes prioritaires de ces composants. Les actions se feront toujours au moyen d'émulations.

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

Command Post est compatible avec OSF/DME et donc avec HP OpenView.

GESTION DE DONNÉES.

Nous l'avons déjà dit, un serveur de données fait partie de l'environnement Command/Post. Le SGBD relationnel utilisé pour gérer cette BD est compatible SQL; il s'agit de SYBASE.

INTERFACE UTILISATEUR.

L'interface utilisateur est de type graphique et repose sur OSF/Motif ou OpenLook. Sept types d'écrans sont disponibles et peuvent être personnalisés par les utilisateurs. Il y a deux affichages sous forme graphique (des cartes et des graphiques) et cinq affichages textuels.

Un seul de ceux-ci (Active Alert Display) est à la base du traitement des problèmes signalés par les alarmes; les autres ne servent qu'à afficher les informations contenues dans les bases de données sous formes de rapports et de résumés. Il est ainsi possible de procéder à des interrogations basées sur un filtre construit par l'opérateur,

* Restructured eXtended eXecutor : langage développé par IBM pour l'automation de tâches.

d'afficher les 120 derniers évènements (émanant des ressources gérées ou résultant de l'action des opérateurs), ou un résumé des informations contenues dans la BD.

Les cartes peuvent certes représenter des cartes géographiques mais peuvent également être des représentations de l'architecture logique. Ces affichages se basent sur des dessins ou des images scannées. Des possibilités de zoom sont offertes, mais cet affichage ne peut servir de point de départ pour la prise en charge d'une alarme.

Il faut noter que les vues sont propres à celui qui les visualise et pas nécessairement communes à tous les utilisateurs.

FONCTIONNALITÉS.

D'emblée, disons que Command/Post est prévu pour gérer les fautes (sous quelques réserves) et faire du "reporting" et de la gestion de performances. Si l'on désire faire autre chose, il faudra nécessairement passer par des émulateurs qui feront perdre aux opérateurs le bénéfice de disposer d'un intégrateur.

En ce qui concerne la gestion intégrée des fautes, il est conseillé de recourir à l'utilisation d'Auto-Command, faute de quoi on devra utiliser des émulateurs en ligne. Command/Post est avant tout un outil permettant une meilleure visualisation des alarmes générées par les diverses ressources gérées ainsi qu'un point d'accès centralisé, mais pas intégré, aux divers systèmes.

Etant donné que Command/Post enregistre tous les évènements dans des bases de données relationnelles SQL, il est possible de générer des rapports à partir d'un générateur de rapport compatible SQL ou au moyen de Sybase's Data Work Bench. Notons que les informations stockées ne sont que des évènements émanant des ressources ou des opérateurs et non des mesures.

Il n'est fait mention que de deux types d'utilisateurs: les opérateurs et les superviseurs. Le superviseur gère les utilisateurs et peut ainsi créer des distinctions parmi ceux-ci.

DIVERS.

Fonctionnement : tout système de gestion d'éléments émet des informations. Avant d'être utilisées par Command/Post, ces informations doivent faire l'objet d'un traitement. Durant celui-ci, on peut ajouter des informations (par exemple : des renseignements concernant la connectivité ajoutés sur base de l'identifiant de ligne qui est la clé d'accès à une BD); ce type de traitement est fait au moyen de transformateurs. Les messages ainsi transformés peuvent alors passer au travers du moteur de filtrage (ALFE - Alert Logic Filter Editor); il deviendra alors une alerte au sens de Command/Post. Ce deuxième traitement est le coeur de Command/Post; c'est lui qui va entraîner l'affichage du message sous une forme plus explicite.

Pour pouvoir procéder à ce second traitement, on aura dû définir au préalable et pour chaque message, la règle de filtrage associée. Tout message sélectionné fait alors l'objet d'une spécification de l'alerte associée. C'est cette alerte ainsi définie qui sera à la base de l'affichage à l'écran lorsqu'un système de gestion d'éléments émettra une information, un message. Lors de l'apparition d'une telle alerte sur son écran, l'opérateur qui désire la traiter commencera par se l'assigner; ensuite, il pourra obtenir plus d'informations à son sujet ou redéfinir les éléments affichés. Enfin, s'il dispose d'Auto-Command, il pourra déclencher le scénario prévu s'il n'est pas automatique, sinon il verra s'afficher la prise en charge par l'opérateur automatisé.

AVANTAGES ET INCONVÉNIENTS.

☺ Facilité de personnalisation des informations affichées.

☺ Possibilités de reporting offertes.

☺ Filtrage des alarmes facilement paramétrable.

☺ Simplicité de fonctionnement, d'où une grande capacité de traitement des messages.

☺ Capacité d'automatisation.

☺ Capacité potentielle à recevoir des messages de n'importe quel appareil (détecteur incendie, climatisation, ...) .

☹ Utilisation de REXX pour créer les scripts.

☹ Simplicité de fonctionnement, d'où une faible valeur ajoutée.

☹ Absence de système expert.

☹ Actions possibles via émulateurs.

3.1.3. BULL ISM (Integrated System Management).

ARCHITECTURE.

ISM fonctionne sous UNIX.

ISM est un système intégré de gestion de système distribué.

La station de gestion d'ISM a une architecture interne basée sur l'échange d'informations de gestion à l'aide du service CMIS et autorise l'utilisation de multiples protocoles d'administration (SNMP, CMIP, ...). Un ensemble d'applications de gestion peut venir se greffer sur le système central. Le coeur d'ISM, ce sont les *Services Communs* basés sur CMIS. On y retrouve : les services de journalisation (logging) des alarmes et évènements, le service d'accès aux *descriptions de classes d'objets* (MIB Template), le service d'accès à des données persistantes par des commandes CMIS et enfin le routage des commandes et évènements vers les gestionnaires d'objets, de service et vers les applications. On retrouve également un compilateur d'objets servant à la définition des objets utilisables et au stockage des informations pouvant être utilisées par les outils. La communication avec les agents se fait par l'intermédiare *d'Intégrateurs d'Agents* (Agent Integrator). Enfin, le dernier composant important est un interpréteur ISM-ML qui comporte des librairies chargées d'interfacer avec les services nécessaires aux applications développées dans cet environnement. ISM/ML est le langage utilisé pour le développement d'applications d'administration. Ce langage permet, entre autres, l'accès aux services CMIS, aux descriptions de classes d'objets, l'accès SQL et les communications inter-applications (permet d'exporter des fonctions génériques faisant partie du noyau d'ISM).

ISM peut s'intégrer dans une organisation hiérarchique de gestionnaires et peut dans ce cas servir de gestionnaire de gestionnaires mais également de gestionnaire local ou spécialisé. Le Manager of Managers gèrera l'activité globale et certaines alarmes importantes prédéfinies.

DOMAINE(S).

ISM permet la gestion de:

- LAN reposant sur Novell Netware 3.11 et 3.12 ou sur LAN manager for UNIX,
- WAN (SNA, X.25 et DSA),
- grands systèmes (Bull MVS, Bull GCOS 7 et 8),
- exploitations UNIX,
- applications (par exemple : système de messagerie reposant sur X.400).

EXTENSIBILITÉ.

ISM posséde un environnement de développement permettant l'ajout de nouveaux objets ainsi que la définition d'agents SNMP et CMIP, le développement de gestionnaires d'objets et le développement d'applications de gestion.

Les applications ajoutées au système peuvent être écrites en SML (System Management Language), langage propre à ISM, ou en C. L'interface avec le système sera respectivement réalisée au moyen de l'interface SML-CMIS et de l'API SML-MIB Toolkit ou au moyen de l'API XMP.

Le langage SML est un langage de quatrième génération et permet d'accéder à des librairies compilées (boîte à outils graphiques GO, accès SQL, ...). L'interpréteur est événementiel et réagit aux primitives CMIS, aux "timers" et aux évènements graphiques. Les nouvelles applications ainsi créées peuvent être associées aux applications génériques et interactives et permettent la réalisation d'applications spécifiques de supervision homogènes.

SML Interface Builder est un environnement de développement d'interface utilisateur pour les développeurs d'applications SML. Il permet la construction d'interface OSF/Motif. Ses principales caractéristiques sont sa facilité d'utilisation, la rapidité de développement qu'il offre et la possibilité de réutilisation d'interfaces existantes.

INTERFACE (DE GESTION DE) RÉSEAU.

ISM permet de se connecter à des équipements implémentant TCP/IP, SNMP, CMIP, SNA, SMT (liaisons FDDI) et GMP (protocole d'administration pour l'administration des plates-formes GCOS) X.25, IPX/SPX, DSA, Ethernet.

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

ISM est compatible avec OSF/DME.

Il peut servir de gestionnaire de gestionnaires pour tout système de gestion compatible avec OSI/NM Forum. De plus, il est compatible avec certains applicatifs existants (Optivity/Synoptics, Netbuilder/3COM, ...)

GESTION DE DONNÉES.

ISM permet de gérer les BD ORACLE et de visualiser des serveurs et des liens entre les BD, de transmettre des alarmes et de suivre les performances.

Le format des données d'ISM est compatible SQL.

INTERFACE UTILISATEUR.

L'interface utilisateur d'ISM est basée sur X/Motif et permet une utilisation intensive de la technique "pointer et cliquer". Il est possible de personnaliser les affichages au moyen de l'outil SML Interface Builder.

FONCTIONNALITÉS.

Au niveau de la gestion des fautes, ISM permet la visualisation des fautes et des alarmes générées dans les systèmes gérés. Deux types d'automatisation des traitements sont prévus; ce sont les filtres et les scripts. Suivant la complexité de l'action à effectuer, on choisira l'un ou l'autre. Les évènements pourront déclencher des affichages à l'écran, l'envoi de mail et/ou d'un appel sémaphone, ...

La consultation et la modification des configurations des divers composants des réseaux sont possibles à partir d'ISM.

Le "logging" des évènements et des prises de mesures permet la création de tableaux de bord et la visualisation des alarmes en temps réel ou en différé.

L'auto-découverte des composants de LAN et de réseaux X.25 est possible.

La gestion de la comptabilité est possible sur les réseaux X.25. Sur ce même type de réseau, il est possible d'automatiser les actions

Comme la plupart des SIGR, ISM utilise le système de trouble ticket (rapport de problème) produit par Remedy (Action Request System). Ce système peut être paramétré et sert de base de connaissances interrogeable par les opérateurs.

Au point de vue de la sécurité logique, cinq types de fonctions sont assurées: fonctions administratives (utilisateur et application), fonctions d'accueil utilisateur (bureau personnalisé et mot de passe unique), fonctions de contrôle d'accès aux ressources du PC par carte, fonctions d'audit et fonctions de confidentialité des échanges inter-applications.

Outre ces fonctions de gestion de réseau(x), ISM fédère également les fonctions d'administration et d'exploitation suivantes : gestion et exploitation des serveurs applicatifs UNIX (gestion des utilisateurs, des processus, des périphériques et BD), gestion de TUXEDO (environnement transactionnel), gestion d'un serveur de sauvegarde, distribution de logiciels vers les systèmes UNIX à partir d'un serveur, distribution de logiciels vers les postes de travail et la gestion d'inventaire (relevé des instances d'objets et leur configuration, ...)

DIVERS.

AVANTAGES ET INCONVÉNIENTS.

☺ Intégration au niveau de l'entreprise (gestion totale des ressources informatiques).

☺ Scripts simples reposant sur le shell UNIX.

☺ Bonne possibilité d'extension.

⊗ Absence de système expert.

⊗ Langage spécifique de programmation des applications.

3.1.4. CABLETRON - Spectrum.

ARCHITECTURE.

Spectrum tourne sous UNIX et permet de choisir parmi plusieurs constructeurs (Sun, IBM, DEC, HP, Silicon Graphix). Spectrum est une plate-forme de conception orientée objet fournie avec un certain nombre de modules de base.

Spectrum repose sur une architecture client/serveur distribuée. Les clients, c.-à-d. les postes de travail des opérateurs, ne disposent que des modules d'affichage des informations et d'interface avec les utilisateurs, tandis que les serveurs se partagent l'ensemble des fonctions de gestion. Cette répartition des fonctions entre clients et serveurs permet de minimiser le volume du trafic en ne faisant circuler que des informations sans leur présentation (cartes, icônes, ...). La distribution des fonctions de serveur est importante dans le cas de grands réseaux complexes dont la surveillance est répartie entre les différents opérateurs, chacun s'occupant par exemple d'un réseau. Remarquons qu'on ne peut connecter que dix clients à un serveur.

Il est permis d'organiser hiérarchiquement plusieurs systèmes Spectrum; un système sert de gestionnaire pour un réseau multi-vendeurs et communique avec un système Spectrum jouant le rôle de SIGR. Une organisation "Peer-to-Peer" (coopération horizontale) est également possible puisqu'il est prévu une répartition des tâches entre les différents serveurs qui communiquent entre eux.

Il est permis d'assurer la gestion à partir d'une station cliente distante.

Le coeur de Spectrum, c'est IMT (Inductive Modelling Technology): il s'agit d'un composant implémentant une certaine forme d'intelligence artificielle. Il permet la mise en évidence et la prise en compte des relations entre les divers composants, ce qui évite la prise en compte de fausses alarmes générées suite à l'impossibilité de se connecter à un élément situé au-delà du composant défaillant. Il crée lui-même le modèle des réseaux gérés. Il permet le déclenchement automatique des actions à entreprendre en cas d'erreur et il assure une surveillance régulière des systèmes gérés, isolant les problèmes détectés.

Un autre composant important est le DCM (Device Communication Manager): celui-ci assure la traduction des protocoles utilisés par les éléments sous-jacents afin de rendre les informations qu'ils émettent compréhensibles pour Spectrum.

DOMAINE(S).

Spectrum se présente comme un système de gestion des réseaux d'une entreprise. Il offre des possibilités de gestion des LAN, WAN (réseaux SNA et ATM), PABX. Il permet également de gérer des stations de travail et des serveurs.

EXTENSIBILITÉ.

Deux types d'outils sont disponibles: le premier est mis à la disposition des utilisateurs afin qu'ils puissent personnaliser leurs icônes, vues, ... ; le second est destiné aux développeurs et comprend les APIs nécessaires pour interagir avec l'interface graphique et le serveur. Il leur permet ainsi de créer de nouvelles applications et interfaces graphiques.

La collaboration avec divers constructeurs assure un développement permanent et nécessite une certaine facilité d'extension modulaire.

INTERFACE (DE GESTION DE) RÉSEAU.

Actuellement Spectrum est compatible avec SNMP et SNA, mais potentiellement, il n'existe pas de limite à condition d'adapter le DCM.

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

En tant que plate-forme assurant le rôle de Manager of Managers (MoM), Spectrum est compatible avec les systèmes de gestion les plus répandus que sont HP OpenView, SunNet Manager, IBM NetView et Novell NMS.

GESTION DE DONNÉES.

Les données de gestion utilisées par Spectrum peuvent être exportées suivant un des formats suivants : ASCII, SAS et Ingres. Celles-ci peuvent être utilisées pour faire des rapports, des études ou pour toute autre application. Le format ASCII délimité en permettra l'utilisation par la plupart des SGBD SQL. Une interface en ligne permet l'accès direct aux données contenues dans la/les BD du/des serveurs.

INTERFACE UTILISATEUR.

L'interface utilisateur est graphique et semble faire abondamment appel à la technique "pointer et cliquer". Les diverses vues sont adaptées à tout changement en temps réel. Des sondes peuvent être paramétrées simplement en pointant sur des objets, sans programmation. Pour assurer une interface la plus graphique possible, il existe d'office une série d'icônes, de jauges, de symboles qui peuvent être utilisés pour la réalisation des écrans d'affichage. Outre ceux qui sont fournis, il est toujours possible d'en créer de nouveaux et il est même permis d'inclure des images.

Trois types de vues hiérarchiques sont définis et permettent de zoomer afin d'affiner la vue; ce sont les vues de la topologie logique, de la topologie physique (localisation) et de l'organisation (par département, etc). Elles permettent d'afficher respectivement les connexions et le statut des réseaux, la localisation des réseaux dans l'entreprise et l'attribution des équipements aux départements de la compagnie.

Cinq vues de gestion sont également offertes; elles représentent le statut actuel des appareils et les évènements enregistrés, un peu comme une photo des conditions de fonctionnement du/des réseau(x). Il y a tout d'abord la vue d'alarmes qui affiche les alarmes actives accompagnées d'une icône de l'équipement concerné. Il y a la liste des évènements et alarmes issus des réseaux et de Spectrum. Vient ensuite la vue des "objets trouvés" où l'on retrouve les objets découverts par l'IMT mais qui n'ont pas pu être placés dans une autre vue par manque d'informations. Vient encore la vue de réparation dans laquelle on retrouvera les composants défectueux détenus par chaque utilisateur. Enfin, il y a la vue des découvertes qui permet de retrouver des objets ou groupes d'objets dans les réseaux.

Le système est doté d'une possibilité d'auto-découverte qui permet la mise à jour automatique de toutes ces vues; celles ne pouvant être attribuées par ce système seront placées dans la vue des "objets trouvés" et feront l'objet d'une affectation manuelle par l'utilisateur.

FONCTIONNALITÉS.

En ce qui concerne la gestion des fautes et des alarmes, outre leur affichage, Spectrum permet le couplage avec le système de trouble ticket (rapport de fautes) développé par Remedy (Action Request System - ARS). Ce système facilite la détection, le suivi et la résolution de tous les problèmes. Il est également possible d'automatiser le traitement au moyen de scripts écrits par les utilisateurs.

La gestion des configurations est prise en charge par Spectrum. Ainsi, tout objet modélisé peut faire l'objet d'une configuration à partir du SIGR. Une vue des découvertes permet la visualisation et la modification de la configuration des objets; une autre permet l'affichage de leur performances. Il est de plus possible d'observer et de stocker diverses informations au sujet des performances d'un réseau ou partie de réseau; par exemple, nombre de paquets/sec, d'erreurs/sec et de collisions/sec. Ultérieurement, ces données ainsi que toutes celles stockées et exportées vers des BD pourront être utilisées afin de réaliser un rapport. Un module de "reporting" peut être ajouté; il contient six types de rapports prédéfinis contenant des graphiques et pouvant être personnalisés voire même redéfinis.

Il est possible de faire de l'observation de réseaux à distance par délégation (Remote MONitoring), en utilisant dans les LAN des composants intelligents construits par Cabletron.

Spectrum permet la définition de différents profils d'utilisateurs en déterminant, entre autres, les vues et les niveaux d'accès associés.

Il est possible de lancer automatiquement des scripts UNIX ou des applications SPECTRUM (par exemple : reporting) au moyen d'un Scheduler.

DIVERS.

Fonctionnement : les gestionnaires d'éléments ou les composants émettent des informations suivant leur format. Celles-ci passent alors au travers du DCM qui les transforme pour les rendre compréhensibles par le noyau de Spectrum. A ce moment, elles sont analysées, filtrées et éventuellement adressées à une station de travail qui modifiera ses vues en conséquence. Un script pourra être déclenché automatiquement et/ou l'utilisateur lancera une application (Spectrum ou non); au passage, cet évènement aura été stocké dans la base de données du Système.

Installation : initialement, il est possible soit de créer un fichier de configuration, soit de charger la configuration existante dans les systèmes.

AVANTAGES ET INCONVÉNIENTS.

☺ Existence d'un système expert.

☺ Support pour le trouble ticketing.

☺ Prise en charge d'un maximum de gestion.

☺ Compatibilité potentielle avec tous les protocoles.

☺ Interfaçage simplifié et personnalisable.

☺ Compatibilité avec les systèmes de gestion courants.

☹ Maturité et popularité du système

Remarque : le peu de détails que nous possédons sur la programmation du DCM et des autres applications ne nous permet pas de prendre position sur l'ouverture du système.

3.1.5. Digital - *Polycenter TeMIP.*

ARCHITECTURE.

Ce système toune sous UNIX. Son architecture est de type client/serveur et est distribuée. La liaison entre les serveurs et les clients peut être faite par des réseaux de type Ethernet, Token-Ring, ATM, X.25, ISDN et FDDI. L'architecture logique de ce produit se compose d'un "repository" d'informations de gestion, de l'"Executive" qui est le noyau, de modules de présentation chargés de l'interfaçage avec les utilisateurs, de modules de fonctions étendant les fonctionnalités de base et de modules d'accès chargés de l'interfaçage avec les entités gérées.

Il est possible d'organiser TeMIP soit en respectant une communication "Peer-to-Peer" (coopération horizontale) soit hiérarchiquement. Il est également possible de l'intégrer avec d'autres systèmes de gestion tels que NetView.

DOMAINE(S).

Ce SIGR permet de gérer des LAN, des WAN, des réseaux privés de téléphonie et des appareils d'environnements (climatisation, capteurs divers,...).

EXTENSIBILITÉ.

Il est possible, à l'aide de boîtes à outils fournissant des API, de développer de nouveaux modules d'accès, de présentation et de fonctions. Les opérateurs peuvent créer leurs propres applications et les ajouter dans les menus. Les extensions seront développées en C.

INTERFACE (DE GESTION DE) RÉSEAU.

Actuellement, il est possible d'accéder aux divers appareils gérés au moyen des protocoles suivants : SNMP, CMIS/P et tout appareil avec lequel il est possible de communiquer au moyen de chaînes ASCII. Notons que parmi les API, il y a l'API/XMP qui permet l'accès aux services de SNMP et CMIS/P.

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

Cette solution se déclare compatible avec la recommandation M.3010, avec OSF/DME, avec OMNIpoint 1. Elle doit de plus permettre une implémentation respectant les principes de CORBA. TeMIP a été pensé pour s'inscrire dans le cadre de la recommandation M.3010.

GESTION DE DONNÉES.

La base de données des applications est de type SQL (Oracle, Ingres)

INTERFACE UTILISATEUR.

L'interface utilisateur se base sur le standard OSF/Motif et X/Windows. Outre les fonctionnalités offertes par ceux-ci, il est possible de personnaliser les éléments d'interfaçage (écrans, contenu des messages, ...). Il est donc possible de représenter les ressources gérées sur une carte. Les vues des réseaux peuvent être personnalisées, par exemple, afin de donner une représentation géographique des ressources.

FONCTIONNALITÉS.

Polycenter dispose des moyens permettant une auto-découverte de toutes les ressources du réseau supportant TCP/IP. Outre une auto-découverte à la demande, il faut savoir que les réseaux gérés restent en permanence sous la surveillance de cette application. Lors de la découverte d'une modification dans un réseau (ajout ou suppression d'une ressource), les vues sont adaptées. Notons que les autres éléments feront l'objet d'une définition manuelle par les administrateurs.

En ce qui concerne les évènements, il est possible de créer des filtres pour leur affichage à l'écran et/ou leur prise en charge. Tous les évènements ou une partie de ceux-ci sélectionnés par application d'un filtre, peuvent être enregistrés pour, par exemple, permettre leur analyse ultérieure dans un rapport. Il en est de même des commandes des opérateurs.

Les opérateurs peuvent spécifier les divers paramètres de surveillance de valeurs critiques ainsi que la réaction qui y est associée. Un script UNIX ou un programme pourra être déclenché à la réception d'une alarme.

Un système de trouble ticketing propre fait partie des modules de fonctionnalités de base. La base de données contenant les trouble tickets pourra être gérée au moyen d'Oracle ou Ingres

Un système expert existe et peut être ajouté comme module de fonction, mais il semble que cela ait été peu employé jusqu'à présent. La base de connaissances est à créer lors de l'installation et il semble qu'elle s'appuie sur les trouble tickets.

La base de données contenant la configuration des divers éléments des réseaux est continuellemnt mise à jour au fur et à mesure des changements.

On peut définir des profils qui seront attribués par la suite aux divers utilisateurs reconnus. Cette technique permet de limiter et de contrôler les accès aux diverses fonctions.

Il est possible d'avoir à l'écran, en temps réel et sous forme graphique, les statistiques concernant les réseaux. Il est tout aussi faisable de les collecter dans un fichier qui sera exploité plus tard au moyen d'un tableur ou d'un SGBD SQL.

DIVERS.

En fait, Digital offre deux types de produits. Ce sont des systèmes de gestion plus ou moins intégrés et un ensemble de produits plus spécifiques. Dans le premier type, on retrouve des produits tels POLYCENTER NetView et POLYCENTER Network Manager 200/400. POLYCENTER NetView est en fait IBM NetView 6000 auquel Digital a ajouté certaines fonctionnalités pour gérer des éléments qui lui sont propres (DECnetIV, DEC LAN bridges, ...). Il faut savoir qu'il existe un accord entre Digital et IBM qui a pour conséquence que toute amélioration de NetView est simultanément implémentée sur les deux plates-formes. Toutefois, les applications développées par les tiers ne doivent pas nécessairement tourner sur les deux produits, chacun de ceux-ci ayant son propre OS (AIX pour NetView 6000 et OSF/1 pour POLYCENTER NetView). POLYCENTER Network Manager 200/400 tourne sur OpenVMS et c'est en se basant sur l'architecture de ces deux produits que TeMIP a été développé. Ce produit pourrait également être repris dans notre liste de SIGR car il offre également des fonctions telles que l'auto-découverte, la gestion d'alarmes, ... Nous avons préféré choisir TeMIP car, à l'heure actuelle, c'est ce produit qui semble faire l'objet de développements de la part des techniciens de Digital. Jusqu'à présent, c'était la solution présentée par Digital pour la gestion au niveau de l'entreprise. Ce produit n'est pas présenté comme pouvant être organisé en "Peer-to-Peer" ou servir de Manager of Managers (MoM). Il semble que la totalité des équipements doivent être liés à POLYCENTER Network Manager.

ATOUTS ET INCONVÉNIENTS.

☺ Conformité à la recommandation M.3010.

☺ Architecture modulaire.

☺ Souplesse et flexibilité.

☺ Intégration de systèmes anciens et divers grâce au module d'accès ASCII.

☹ Nouveauté relative et faible distribution (peu de modules fonctionnels)

3.1.6. HEWLETT PACKARD - OpenView.

ARCHITECTURE.

OpenView est une gamme de produits basés sur les plates-formes de gestion du même nom. Deux plates-formes existent: la première est simple tandis que la seconde est distribuée. Toutes deux tournent sous UNIX; une version sous Windows existe mais est nettement moins populaire. Leur architecture générale est composée de:

- *Objets Gérés* (Managed Objects) qui sont une abstraction des ressources réelles gérées.
- *Services Communs de Gestion* (Common Management Services) qui jouent un rôle de pont entre l'infrastructure de communication de gestion de réseau(x) et les objets gérés. Divers éléments concourent à réaliser ces services : les protocoles de communication, les services de filtrage et de propagation des évènements, la gestion de données avec accès SQL, ...
- *Applications de gestion* (Management Applications) qui offrent les diverses fonctionnalités de gestion.
- *Interface Utilisateur Commune* (Common User Interface) qui offre ainsi une intégration de la présentation quelle que soit l'application sous-jacente.

DOMAINE(S).

Bien qu'il s'agisse d'une plate-forme basée sur l'architecture OSI et qui a été optimisée pour inclure les réseaux TCP/IP, elle est essentiellement tournée vers la gestion de ressources supportant SNMP. De ce fait, nous pensons qu'elle est principalement axée sur les LAN et, dans une certaine mesure, vers les WAN. Toutefois, il n'existe potentiellement aucune limite puisque ce système permet l'adjonction de MIB spécifiques. Il est utile de préciser que outre les télécommunications, HP OpenView se veut également être une plate-forme pouvant intégrer la gestion de systèmes.

EXTENSIBILITÉ.

Diverses API sont prévues et permettent l'intégration de diverses applications dans cet ensemble. De très nombreux produits développés par des tiers ou par HP sont actuellement disponibles. Ce type de produit est modulaire, ce qui facilite son extensibilité.

INTERFACE (DE GESTION DE) RÉSEAU.

Initialement prévu pour gérer des ressources implémentant SNMP, il peut également surveiller tout composant doté d'une adresse IP. Toutefois, étant donné la disponibilité de l'XMP/API, il est possible de gérer des ressources CMIP.

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

HP fait partie de l'association OSF et, de ce fait, leur produit est compatible avec OSF/DME.

GESTION DE DONNÉES.

Les données sont stockées sous le format ASCII, mais il est également possible de disposer d'un outil de gestion de base de données de type relationnel, compatible avec Ingres.

INTERFACE UTILISATEUR.

L'interface utilisateur repose sur OSF/Motif, mais HP a développé son propre système d'interface afin d'offrir une intégration à ce niveau. Etant donné les nombreux modules développés par les tiers, il était nécessaire d'offrir une interface commune permettant ainsi de garder une transparence totale pour l'utilisateur. Elle s'appelle HP OpenView Windows.

C'est sur base d'une carte que les diverses ressources sont présentées à l'utilisateur. Cette dernière reprend la topologie générale et reconnaît l'organisation en bus, en anneau à jeton, en étoile et en anneau FDDI. Cette carte reprend deux types de symboles paramétrables et personnalisables. On retrouvera ainsi des icônes représentant des appareils et des symboles de connexion; ces derniers relieront deux icônes ou encore une icône et un "backbone". Le nombre de symboles n'est pas limité; chaque application et/ou utilisateur peut en ajouter. L'utilisateur peut ajouter des symboles de ressources qui n'auraient pas été découvertes automatiquement. Une hiérarchie des cartes peut être construite par l'utilisateur qui pourra également créer des vues spécifiques à ses besoins. Chacune de ces cartes pourra se voir dotée d'un fond sous forme d'une carte, d'un plan ou d'une photo au format GIF ou X11.

Lors de la survenance d'un évènement, la couleur du symbole sera modifiée afin de renseigner un changement de statut. Des menus contextuels permettent d'obtenir de plus amples renseignements sur tout symbole

sélectionné; de plus, chaque symbole peut être extensible, c.-à-d. qu'il ouvrira une nouvelle vue lors d'un double click, ou exécutable, c.-à-d. que tout double click sur celui-ci entraînera le déclenchement d'une action. Il faut encore dire qu'il ne semble y avoir aucune possibilité de visualiser une alarme sous forme de texte.

La barre de menu reste standard quelles que soient les applications ajoutées; le menu déroulant attaché à chaque item de celle-ci sera adapté afin d'intégrer la nouvelle application disponible et directement exécutable.

FONCTIONNALITÉS.

Nous nous limiterons ici à parler des fonctionnalités de la plate-forme en elle-même; il est évident qu'il est possible de couvrir tous les domaines de la gestion suivant l'ISO en ajoutant diverses applications développées par des tiers ou en en développant soi-même.

Parmi les composants de base, il y en a un qui offre des services d'auto-découverte et d'affichage de toutes les ressources disposant d'une adresse IP. L'affichage se fait au moyen de HP OpenView Windows et de ses fonctionnalités de cartographie et de gestion de symboles. Cette auto-découverte peut être limitée à certains éléments repris dans un fichier ASCII (*seed file*). Elle peut également être complétée manuellement.

Un autre composant offre des services de gestion des évènements; il reçoit les évènements, les filtre et les redistribue vers les applications qui en ont fait la demande. De même il pourra les conserver en mémoire.

Le *MIB Loader/Browser* permet de configurer la MIB en ajoutant à la MIB standard certaines MIB spécifiques ou même en permettant la définition de nouveaux éléments. Cet élément permet également de consulter ou d'affecter une valeur d'objet de la MIB.

Les outils de présentation de données (Data Presentation Tools) qui font partie de la plate-forme de base permettent une présentation graphique des données à partir d'un fichier en temps réel ou a posteriori. Ils peuvent également servir d'interface graphique générique.

DIVERS.

HP OpenView Network Node Manager est un des produits de la gamme qui est le plus populaire; ses fonctionnalités sont très proches de celles offertes par IBM avec AIX NetView/6000. Ceci est logique lorsqu'on sait que IBM s'est basé sur HP OpenView. La gestion de la configuration et des performances est relativement bien couverte. La gestion des fautes est possible mais on préfèrera utiliser l'application OperationsCenter.

HP OpenView OperationsCenter est un produit répondant aux besoins de centres de contrôle. Il offre la centralisation de toutes les alarmes et la mise sous un format commun, la surveillance avec le placement de limites, le stockage des divers messages et le déclenchement d'actions automatiques ou non. Cette application offre donc essentiellement des moyens pour faire de la gestion de fautes et de performances.

AVANTAGES ET INCONVÉNIENTS.

☺ Popularité.

☺ Modularité.

⊗ Fonctionnalités de la plate-forme.

3.1.7. IBM -AIX NetView/6000.

ARCHITECTURE.

Ce système est prévu pour tourner sous UNIX (AIX de chez IBM). Une distribution des fonctions de gestion est possible en utilisant Systems Monitor/6000 pour la gestion à distance de sous-réseaux, par exemple. AIX NetView/6000 joue alors un rôle de superviseur.

Il peut s'intégrer dans l'architecture ONA; sa connexion avec le focal point est alors établie au moyen de AIX NetView Service Point program. Cette organisation permet une gestion coopérative entre des environnements SNA et TCP/IP.

D'après nos informations et bien que cela ne soit pas spécifié dans la documentation du constructeur, il semble que AIX NetView/6000 a été développé sur base de la plate-forme HP OpenView; cela n'empêche pas ce système de se présenter comme une plate-forme ouverte de gestion de réseau(x) et de système permettant une gestion centrale et distribuée.

DOMAINE(S).

Le domaine privilégié est celui des équipements reposant sur la suite de protocoles TCP/IP. Cela signifie aussi bien les WAN que les LAN. Avec le système seul, il n'est toutefois possible que de gérer des ressources implémentant SNMP. Une API est disponible afin de permettre l'intégration d'applications de gestion suivant le protocole CMIP. Ici encore, il est possible d'intégrer des applications de gestion de tout environnement, mais elles ne font pas partie intégrante du système de base.

EXTENSIBILITÉ.

Comme toute plate-forme et par définition, ce produit se veut extensible puisqu'il doit servir de base au développement d'applications de gestion par des tiers. Ainsi, un certain nombre d'applications existent déjà, mais la popularité limitée de ce système, notamment auprès des développeurs, fait que le catalogue n'est pas aussi important que celui de HP OpenView.

Plusieurs API sont fournies et/ou disponibles. Ainsi retrouve-t-on, entre autres, l'API XMP qui supporte la gestion de ressources au moyen de SNMP et de CMIP, la End-User Interface API qui permet l'intégration d'applications dans l'affichage graphique, l'Event Filtering API qui autorise la définition des filtres et limites pour la transmission d'évènements vers les applications et l'API SNMP qui permet l'accès à des informations contenues dans la MIB.

INTERFACE (DE GESTION DE) RÉSEAU.

Le système AIX NetView/6000 est avant tout un système de gestion de ressources SNMP. Toutefois, la disponibilité de l'API-XMP permet également l'interaction d'applications avec des ressources suivant CMIP.

Outre TCP et IP et leur pendant dans le modèle OSI, AIX NetView/6000 reconnaît les protocoles DECnet, Frame Relay et les appareils communiquant au moyen de flux de caractères, d'interface RS-232 et d'interface parallèle comme les imprimantes. Il peut communiquer avec des équipements suivant AppleTalk et le RMON (Remote MONitoring - surveillance à distance) est également possible.

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

Tout comme HP OpenView, sur base duquel il a été développé, il est compatible avec OSF/DME.

GESTION DE DONNÉES.

Les données décrivant les ressources gérées sont sauvegardées dans une base de données relationnelle de type Ingres. Les autres données sont stockées sous forme de fichiers ASCII et peuvent être utilisées ultérieurement, par exemple, dans un tableur.

INTERFACE UTILISATEUR.

L'interface utilisateur est un des points qui ont été développés par rapport à HP OpenView. Ici encore, elle est basée sur OSF/Motif et facilite la surveillance dynamique des réseaux. L'écran standard est composé de quatre fenêtres.

- Il y a tout d'abord la *vue topologique* (comme dans HP OpenView) qui permet l'affichage des réseaux sous forme de cartes. Ces cartes peuvent représenter aussi bien des vues physiques que des vues logiques. Elles sont dynamiques et organisées hiérarchiquement, c.-à-d. que ces vues sont les représentations graphiques des réseaux à différents niveaux (par exemple : réseau, segment, noeud). En outre, il est possible d'organiser plusieurs hiérarchies (par exemple suivant le protocole sous-jacent); bien entendu chacune de ces cartes

pourra faire l'objet d'une personnalisation (figure de fond, icônes, ...). Notons enfin que plusieurs topologies physiques sont reconnues; ce sont les topologies en étoile, en anneau, en arbre et en bus.

- Ensuite il y a *l'arbre de navigation* qui permet à l'opérateur de savoir en permanence où il se trouve dans la hiérarchie des vues. Ainsi, à chaque fois qu'une nouvelle vue est ouverte, son icône apparaît dans l'arbre ainsi que les relations qu'elle a avec les autres vues.
- Il y a encore le *bureau de contrôle* où apparaîtront les alarmes dans une liste ou sous forme de fiches et à partir d'où elles seront gérées. On y retrouvera également les applications associées; par exemple, une application nécessaire pour la recherche des causes d'erreurs.
- Enfin, il y la *palette d'outils* qui reprend sous forme d'icônes tous les outils couramment utilisés et qui peuvent ainsi être directement lancés.

Tout évènement se signalera sur la carte au moyen d'une modification de la couleur de l'icône. Des règles de propagation des évènements dans les vues hiérarchiquement supérieures existent et sont personnalisables. De plus, l'évènement est décrit sous forme textuelle dans une liste ou dans une fiche. Cette dernière pourra en outre se voir complétée par divers renseignements comme par exemple, un extrait de la MIB de la ressource concernée et/ou des commentaires de la part de l'opérateur.

Enfin, la barre de menu et d'éventuels menus contextuels rendent possible l'exécution de toute opération; il n'est donc pas nécessaire de recourir à des commandes en ligne.

FONCTIONNALITÉS.

L'auto-découverte est bien entendu ici encore possible, mais contrairement à HP OpenView elle n'est pas limitée aux ressources disposant d'une adresse IP. En effet, elle supporte les applications nécessaires à l'intégration et à la corrélation d'autres topologies et permet de s'en servir tant pour le dessin des cartes que pour l'affichage des évènements.

En ce qui concerne la configuration, il est possible de consulter et d'affecter les valeurs des objets de la MIB. Des fonctions de "browse" et de "query" sont fournies. De plus, il faut savoir qu'il est possible de construire, au moyen d'une interface et sans programmation à proprement parler, des applications de gestion des informations de la MIB. Les informations ainsi collectées pourront être utilisées sous forme de tables ou dans des graphiques soit en temps réel, soit après stockage dans des fichiers.

Ceci nous amène tout naturellement à parler des fonctions de gestion des performances et du "reporting". Toutes les informations récoltées sont paramétrables et pourront faire l'objet d'impression ou de sauvegarde pour la constitution de rapports graphiques ou encore être exploitées dans un tableur. Il faut savoir que de telles informations peuvent également être affichées en temps réel.

En ce qui concerne la gestion des fautes, outre l'affichage sur la carte, toute faute fera l'objet d'un affichage au moyen d'une fiche. Diverses recherches peuvent être faites parmi ces fiches; par exemple, il est possible de rechercher toutes les fiches portant sur un type d'appareil ayant un même type d'erreur. Il est possible de placer différents niveaux pour le déclenchement d'alarme avant la survenance d'un problème. De même, suite à une alarme, il est possible de paramétrer une réaction automatique entraînant l'exécution d'un "shell script" ou le lancement d'une application.

Enfin, signalons l'existence de trois outils pour diagnostiquer la cause d'une alarme. Ce sont le test IP (= PING), le test de TCP (= établissement d'une connexion TCP) et le test SNMP qui cherche à déterminer la présence d'un agent SNMP. Notons qu'il est possible d'effectuer un PING à distance (entre deux entités distinctes de la station de gestion) et que l'on peut suivre la route suivie par un message.

DIVERS.

Digital, qui propose POLYCENTER NetView, a passé un accord avec IBM afin que toute nouvelle version soit commune aux deux produits.

AVANTAGES ET INCONVÉNIENTS.

☺ Multi-protocoles.

☺ Popularité.

☺ Qualité de l'interface.

☺ Nombreuses API.

☹ Moins d'applications disponibles que pour HP OpenView.

3.1.8. Sun Connect - Solstice SunNet Manager 2.2.2.

ARCHITECTURE.

Ce système tourne sous UNIX (Solaris 1.1.2. et 2.4); il s'agit d'un système client/serveur distribué. Les clients et les serveurs sont reliés par un réseau fonctionnant sous TCP/IP. Construit suivant les principes orientés objet, il repose sur le schéma agent/gestionnaire de OSI. Au coeur du système on retrouve une application appelée *console*. C'est l'endroit à partir duquel les tâches sont exécutées et où les informations issues des requêtes sont retournées. Elles pourront être stockées dans le *Data Log*. On retrouve également:

- la *Management Data Base* (MDB). Elle contient les définitions des types d'éléments gérés, des instances des types d'éléments, des agents créés pour leur gestion et des requêtes pouvant leur être adressées. Il faut remarquer que chaque instance de console utilisera sa propre *RunTime Data Base* (RTDB). Celle-ci contiendra les données de la Management Data Base et les personnalisations faites par chaque instance de *console*. Ces RTDB peuvent être sauvegardées sous forme de fichiers ASCII pour être réutilisées ultérieurement.
- Le *Discover Tool*. Cet outil facilite la constitution de la MDB en découvrant tous les équipements adressables par IP et SNMP. Il élaborera une représentation graphique des éléments ainsi trouvés.
- Le *Results Browser*. Il analyse les données et les évènements, permettant ainsi leur stockage dans le but de pouvoir créer des rapports (textuels).
- Le *Results Grapher*. Il permet d'avoir une représentation graphique des résultats produits par le *Results Browser* ou directement issus des "data" ou "event log". Il permet l'affichage en direct de toute donnée résultant d'une requête.
- Le *Request Builder*. Il facilite la construction rapide de requêtes d'évènements (surveillance par rapport à un niveau) et de données (recueil permanent à intervalle régulier des valeurs de certains attributs).
- Le *Request Viewer*. Il rend la consultation et la sélection des requêtes disponibles plus aisées en permettant leur affichage trié suivant divers critères; par exemple, sur base des attributs concernés.
- Le *Set Tool*. Cet outil est utilisé pour modifier la configuration; il agit sur les attributs.

Outre ces outils et bases de données, il y a également deux démons: L'*Activity Daemon* et l'*Event Dispatcher*.

- L'*Activity Daemon* garantit que toutes les requêtes lancées et non terminées restent actives et qu'elles continuent à être servies.
- L'*Event Dispatcher* reçoit les évènements et les distribue aux destinataires enregistrés. Il pourra ainsi les stocker, après filtrage, dans l'"*Event Log*" et/ou les envoyer à la console ou encore à toute autre application qui lui en a fait la demande.

Notons que plusieurs requêtes prédéfinies sont disponibles, ce qui facilite le travail des opérateurs qui sont ainsi libérés de la nécessité d'en connaître la syntaxe.

Au point de vue de l'architecture inter-systèmes de gestion, SNM peut être placé sous le contrôle de IBM NetView, c.-à-d. qu'il peut lui transmettre les alertes et recevoir ses commandes.

DOMAINE(S).

SunNet Manager (SNM) est prévu pour gérer des LAN et des WAN. Il semble que la gestion des MAN soit aussi possible. En fait il n'existe pas de limites à conditions que les éléments gérés soient compatibles SNMP ou qu'on écrive un proxy à cette fin.

EXTENSIBILITÉ.

Divers API sont fournis avec SNM et permettent le développement de nouveaux agents et proxys. De plus, il est possible d'écrire de nouvelles applications, par exemple en C ou C++, qui seront intégrées à SNM. Rappelons enfin qu'il s'agit d'une architecture orientée objet, ce qui est un gage de flexibilité. Trois types d'API existent: l'API des services du gestionnaire qui permet l'intégration d'applications propriétaires, l'API des services de l'agent qui permet la communication avec des entités d'un protocole non défini et l'API de la base de données et de la topologie qui permet de modifier la BD et l'affichage de la topologie.

INTERFACE (DE GESTION DE) RÉSEAU.

Les protocoles de gestion de réseau(x) reconnus par SNM sont principalement SNMP et SNMPv2. Il faut savoir qu'il est avant tout prévu pour gérer des éléments compatibles avec SNMP. Il semblerait que CMIS/P soit

également accepté. Divers agents et proxys ont déjà été développés afin de permettre la gestion de réseau(x) fonctionnant sous X.25, SNA, DECNet, ATM, Ethernet et FDDI. Le développement de nouveaux proxys rend possible la communication avec tous les protocoles.

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

Ce SIGR respecte le standard OMNIPoint 1.

GESTION DE DONNÉES.

Tous les fichiers sont de type ASCII; certains outils ont été développés pour les utiliser en SQL. Aucune fonctionnalité de gestion de réseau(x) n'est prévue dans SNM.

INTERFACE UTILISATEUR.

L'interface graphique est basée sur Open Look qui permet la manipulation de fenêtres, boutons, ... Nous avons vu que le *Discover Tool* crée automatiquement une "carte" des éléments qu'il trouve. Cette carte peut être complétée manuellement par l'opérateur; il peut également la personnaliser en redéfinissant les icônes associées aux types d'éléments gérés. Il est d'ailleurs possible de procéder à la définition de ceux-ci, s'ils n'existent pas. Les diverses vues peuvent être disposées sur des cartes, plans ou photos et être organisées hiérarchiquement. Une partie des composants peuvent aussi être regroupés dans des vues particulières, par exemple une vue reprenant tous les routeurs.

Les réactions associées aux évènements peuvent être définies et totalement paramétrées. Il est ainsi possible de déclencher des applications ou des "shells" UNIX.

Pour la personnalisation des divers éléments (définition de filtre, type d'éléments, ...), il est abondamment fait appel à une interface de haut niveau (pointer et cliquer).

FONCTIONNALITÉS.

Du point de vue de SNM au sens strict, il n'existe essentiellement que des possibilités limitées de gestion des fautes et des performances, c.-à-d. l'établissement de rapports ainsi que la surveillance des passages de limites et la prise de mesures en temps réel.

En ce qui concerne la gestion des fautes et problèmes, outre la réception et l'affichage de message(s) lors d'évènements ou en réponse à une demande, il n'est rien prévu. Il semble qu'aucune corrélation ou analyse n'existe; seuls les filtres permettent d'éviter une surcharge.

La fonctionnalité la mieux remplie nous semble être la gestion des performances. Il est en effet possible de stocker bon nombre de données et d'évènements qui plus tard seront utilisés afin de réaliser des rapports chiffrés et/ou graphiques.

Le seul moyen d'action qui est prévu est le Set Tool.

La gestion de ce système prévoit deux types d'utilisateurs : le(s) gestionnaire(s) et les opérateurs. Parmi ces derniers, il est possible de distinguer cinq profils se caractérisant par les droits de lecture et d'écriture qu'ils possèdent. Le mécanisme est assez proche de celui que nous avions trouvé dans CMIS/P. En effet, chaque profil est d'un niveau de sécurité donné (allant de 1 à 5). Chaque objet possède également un niveau d'accès en lecture et un en écriture.

Notons qu'à l'inverse de beaucoup de systèmes de gestion qui ne s'intéressent qu'aux appareils, tous les composants (lignes et appareils) sont ici pris en considération par le système de gestion.

A notre sens, ce produit est essentiellement une base de travail disponible pour les tiers afin d'implémenter diverses applications dans un ensemble unique et cohérent. Effectivement, un grand nombre d'applications développées par des tiers existent et permettent de remplir les cinq domaines de gestion définis par l'ISO (FCAPS).

DIVERS.

Sun Soft vient de sortir Solstice Enterprise Management. Il s'agit d'un ensemble dont fait partie SNM et dont le rôle est de permettre la gestion de l'ensemble des moyens informatiques au niveau d'une entreprise (les lignes et appareils de communication, les systèmes, les applications et les données).

La formation prévue pour le personnel est de trois jours.

AVANTAGES ET INCONVÉNIENTS

☺ Popularité et donc disponibilité d'applications.

☺ Maturité.

☹ Simple base de travail disposant de peu de fonctionnalités intrinsèques.

☹ En cas d'incompatibilité avec SNMP, obligation de passer par des proxys, voire de les développer.

3.1.9. TELINDUS - TOM.

ARCHITECTURE.

Ce produit est le seul qui peut à la fois tourner sous UNIX, Windows et Windows NT. TOM est en fait la plate-forme de Network Managers (NMC Vision) vendue par Telindus et pour laquelle ont été développés des modules permettant la gestion du matériel Telindus. Son architecture est distribuée et client/serveur dans le cas de l'utilisation de Multi-Vision. Ce produit est constitué de plusieurs processus qui communiquent entre eux. Certains forment le noyau central tandis que d'autres sont des modules additionnels. Le coeur du noyau est constitué du *Request Broker* (intermédiaire de traitement des requêtes). Autour de celui-ci on retrouve des processus permettant de remplir les fonctions de gestion. Parmi les applications extérieures au noyau, il y a les *sous-systèmes de protocole* qui permettent l'accès aux appareils gérés par ce protocole. On retrouve également des modules spécifiques à un produit ou à une gamme de produits (PSM - Product Specific Module) qui offrent les fonctionnalités nécessaires à leur gestion. Le noyau est composé d'applications clés telles que le système de représentation cartographique, le manipulateur de performances, le manipulateur de fautes, les sous-systèmes de protocoles, la banque de données et ses outils.

Le *Request Broker* manipule tous les messages qui sont envoyés entre les processus. De ce fait, il est actif en permanence. Il connaît toutes les applications qui fonctionnent, ainsi que les informations qu'elles fournissent et celles qu'elles requièrent. Il renverra aussi les réponses au processus demandeur.

DOMAINE(S).

Tant les LAN que les WAN peuvent aisément être gérés par ce système qui dispose également de modules de gestion pour certains réseaux téléphoniques privés. Il faut noter que Microsoft et Network Managers ont passé des accords afin de permettre l'intégration du gestionnaire de système proposé par Microsoft (Microsoft System Management - MSM) dans NMC Vision; ce dernier devient ainsi un produit de gestion au niveau de l'entreprise.

EXTENSIBILITÉ.

L'architecture de TOM rend chaque application indépendante des autres; lors de son installation, elle se fera connaître du *Request Broker* en lui communiquant les services qu'elle peut fournir et les informations qu'elle attend. Il est possible d'ajouter de nouveaux PSM et PIM (Product Integrator Module - Module d'intégration de produit). Ils auront été développés par les constructeurs de composants de réseaux afin de permettre la gestion de leurs produits; il est également possible de développer ses propres PSM et PIM au moyen de boîtes à outils. Des *sous-systèmes de protocole* peuvent également venir s'ajouter sur le noyau.

INTERFACE (DE GESTION DE) RÉSEAU.

Les réseaux peuvent être gérés en utilisant SNMP, CMIP, IPX, X.25 et MAP/TOP ou encore certains protocoles propriétaires. Les PSM permettent d'obtenir la compatibilité avec d'autres protocoles utilisés par certains appareils.

COMPATIBILITÉ AVEC D'AUTRES STANDARDS.

NMC Vision 3.0 respecte OSF/DME.

GESTION DE DONNÉES.

La base de données de TOM est organisée au moyen de RAIMA Corporation's Data Manager, plus précisément db_VISTA. Ce système se présente comme une base de données rapide basée sur un modèle hiérarchique. Elle est composée de trois parties: les bases de données d'administration, de configuration et de performance. La base de données d'administration contient l'information à propos des divers sous-systèmes et appareils définis. La base de données de configuration décrit la structure et l'interconnnectivité des éléments physiques, ainsi que leurs caractéristiques. Enfin, la base de données des performances contient l'historique des informations de performance des éléments repris dans la base de données de configuration.

Les outils de gestion de base de données sont fournis; il est ainsi possible de supprimer certaines données historiques ou encore d'en sélectionner sur base d'un critère. L'importation et l'exportation de fichiers ASCII est également prévue. Les informations de performance peuvent être importées dans des applications de tiers (tableur ou base de données SQL).

INTERFACE UTILISATEUR.

Basée sur OSF/Motif, l'interface utilisateur fait abondamment utilisation de fenêtres, icônes, menus et de la souris. Lorsque des informations supplémentaires sont nécessaires, elles seront demandées dans une boîte de dialogue.

L'interface avec l'utilisateur se fait au moyen de la carte des réseaux qui est par conséquent active en permanence. Il s'agit en fait de la représentation de toutes les informations contenues dans la base de données constituée par l'administrateur. Une organisation hiérarchique minimale est imposée pour définir les éléments des réseaux. Les opérateurs peuvent personnaliser les cartes et même en créer de nouvelles; elles doivent rester cohérentes avec le modèle spécifié par l'administrateur. Il est ainsi possible de disposer de cartes propres à l'organisation de la gestion des réseaux au sein de l'entreprise. Il va de soi que les icônes pourront également être personnalisées. Une autre caractéristique intéressante de l'interface est la possibilité de visualiser l'appareil géré au moyen d'un dessin fourni par le PSM. Dans certains cas, ce dessin permettra l'interaction avec l'opérateur comme s'il était physiquement présent à coté de l'élément géré.

La *fenêtre de statut* est une autre fenêtre présente en permanence qui présente sous forme graphique la survenance d'un problème un peu à la manière d'un tableau de bord. Les alarmes sont également apparentes sur la carte et il est également possible d'y attacher un signal sonore.

La version 3.0 permet la visualisation de la hiérarchie d'un réseau sous forme d'un arbre (réseau, ensemble des sous-réseaux régionaux, sous réseaux et composants du sous-réseau). Elle permet également de visualiser, au moyen d'un double click, les fautes actives pour un appareil donné. De la même façon, il est encore possible d'ouvrir un mémo associé à un appareil déterminé.

Les lignes physiques sont également gérées et il est permis de représenter, par exemple, deux lignes (l'une active et l'autre de back-up) entre deux équipements. Leur statut sera également représenté au moyen de couleurs.

Désormais, il est possible de lancer des applications non-NMC Vision en même temps que d'appeler le PSM attaché à une icône à partir d'un double click sur celle-ci.

FONCTIONNALITÉS.

La base de travail offre des fonctionnalités permettant d'assurer la gestion des fautes, de la configuration, des performances et de la sécurité.

Dans le domaine de la gestion des fautes, il est permis de définir les différentes alarmes. Cela signifie qu'on leur associe un niveau de gravité (il en existe 5) et éventuellement certains processus à exécuter. Un système de filtrage existe et évitera les avalanches. Un système de trouble ticket propre existe et permet de garder trace des actions exécutées pour résoudre les problèmes. Il est possible de stocker tous les évènements et les fautes dans des fichiers.

Dans le cadre de la gestion de la configuration, il est permis de recueillir et d'affecter les divers paramètres relatifs aux appareils, lignes et sous-réseaux. Il est également possible d'avoir des informations au sujet d'une connexion à partir d'un port particulier. Un système d'aide en ligne est disponible et un contrôle de validité des affectations est fait lors d'interventions sur un élément.

Les fonctionnalités de gestion des performances sont, entre autres, la définition des statistiques à recueillir ainsi que les caractéristiques de cette collecte. Tous les renseignements recueillis peuvent faire l'objet d'une sauvegarde dans un fichier pour une exploitation ultérieure ou être affichés en temps réel sous forme de graphique.

Enfin, la sécurité du système est assurée par un système de mot de passe et par le fait que toutes les opérations ne sont pas possibles pour tous les utilisateurs.

DIVERS.

Les PSM sont des applications permettant d'assurer la gestion d'appareils ou de famille d'appareils spécifiques. On y retrouve l'implémentation des fonctions de gestion nécessaires à la gestion FCAPS. Celles-ci sont optionnelles et laissées au choix des constructeurs. Seules les fonctions de création, modification et suppression de système physique sont obligatoires pour permettre la représentation sur les cartes.

Les PIM sont plus simples que les PSM et offrent l'accès à toutes les caractéristiques standards de NMC Vision telles que la collecte de statistiques ou la vérification des valeurs par rapport à des limites fixées.

AVANTAGES ET INCONVÉNIENTS.

- ☺ Modularité.
- ☺ Popularité croissante.
- ☺ Qualité de l'interface.
- ☺ Multi-protocoles.

- ☹ Absence d'auto-découverte.
- ☹ Absence de filtrage des évènements.

3.2. Etude détaillée de NMC Vision (version 3.0) de Network Managers Ltd.

Dans ce point, nous allons étudier de plus près le produit de Network Managers Ltd soit NMC Vision. Plusieurs raisons motivent ce choix. Tout d'abord, il semble que ce produit connaisse une popularité croissante. Ensuite, il représente le type même de la plate-forme qui offre des fonctionnalités générales mais qui est incapable de gérer un quelconque système sans l'adjonction d'applications particulières. Enfin, sa modularité permet de le voir comme un gestionnaire d'éléments, de réseau ou même de services.

3.2.1. Positionnement dans la hiérarchie TMN (M.3010).

NMC Vision est une plate-forme qui permet de remplir les fonctions de gestion à trois niveaux suivant les modules qui lui sont adjoints (Figure 3.1). Ainsi, le noyau seul avec un PSM (Product Specific Module) remplira le rôle de gestionnaire des éléments correspondant au PSM. Si on lui ajoute les PSM correspondant à tous les éléments du/des réseau(x) ainsi que ceux de RMON et de gestion distribuée (Multi-vision), on obtient un système (intégré) de gestion de réseau(x). Ces deux configurations peuvent fort bien être intégrées dans des systèmes de gestion tels que HP OpenView, NetView/6000 ou SunNet Manager grâce à des kits d'adaptation. Enfin, en intégrant des applications indépendantes, il est possible de remplir les fonctions de gestion de services. C'est le cas si l'on complète NMC Vision avec les produits comme Remedy ARS (fonction de HelpDesk), Isicad Command 5000 (Gestion de liaison - Cable Management), Lynx Layer-8 (Gestion de la comptabilité - Accounting Management), Comdisco (Planning, simulation, conception de réseau) et Microsoft MSM, encore appelé Hermes (Desktop management).

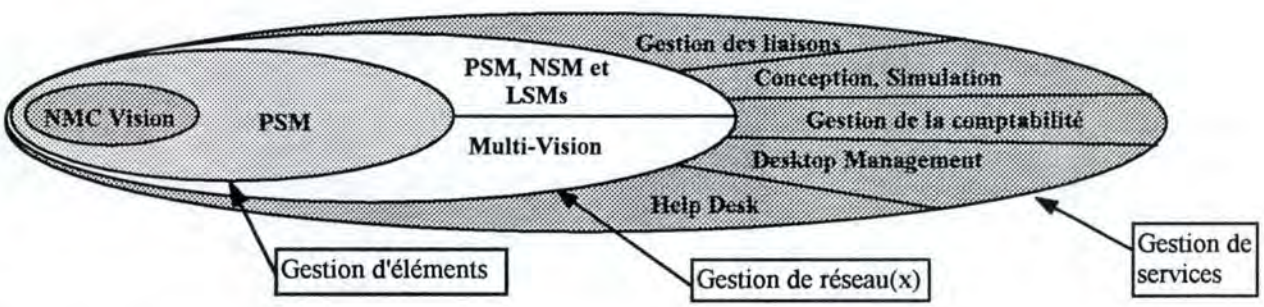


***Figure 3.1.** : NMC Vision et la hiérarchie TMN.*

3.2.2. Architecture détaillée.

La base (Foundation System) de NMC Vision est composée d'un élément central qui est le *Request Broker* (Intermédiaire de traitement des requêtes) et d'applications offrant les fonctionnalités de base du système et dont les processus interagissent (Figure 3.2.). Ces applications sont en quelque sorte des applications génériques qui permettront d'avoir accès aux services propres à chaque produit. Nous pouvons citer :

- le système de carte du/des réseau(x),
- le gestionnaire de performances,
- le gestionnaire de fautes,
- les sous-systèmes de protocole,
- la base de données et ses outils .

Cette organisation permet de remplacer aisément une application par une autre et offre ainsi d'importantes facilités pour évoluer .

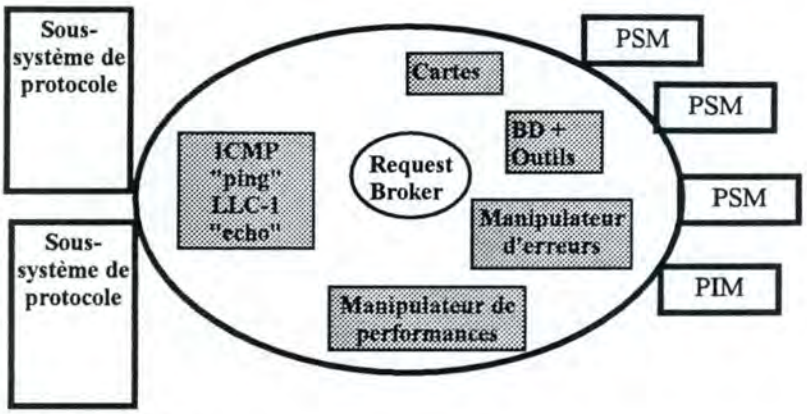


Figure 3.2. : Architecture de NMC Vision.

Sur cette base viennent se greffer des PSM ou des PIM (Product Integrator Module - module d'intégration de produit).

- Le *Request Broker* est le coeur du système. Comme son nom l'indique, il joue le rôle d'intermédiaire entre les divers processus en manipulant tous les messages qui circulent entre ceux-ci. Il est actif en permanence. En fait, c'est comme s'il disposait d'une table reprenant les services que chaque application connectée, c.-à-d. active, peut fournir et ceux qu'elle requière. Cela permet à l'arrivée d'un message de le diriger, un peu à la manière d'un routeur, non seulement vers le processus adéquat, mais également vers l'objet adéquat au sein de celui-ci. Les réponses seront directement renvoyées par le *Request Broker* vers l'application qui a émis la requête.
- Pour l'utilisateur, le *système de cartes du/des réseau(x)*(Figure3.3) est l'élément central. En effet, toute interaction avec le système passe par la carte qui est par conséquent active en permanence. C'est encore sur cette carte et éventuellement au moyen d'un avertissement sonore que les alarmes seront portées à la connaissance de l'opérateur.

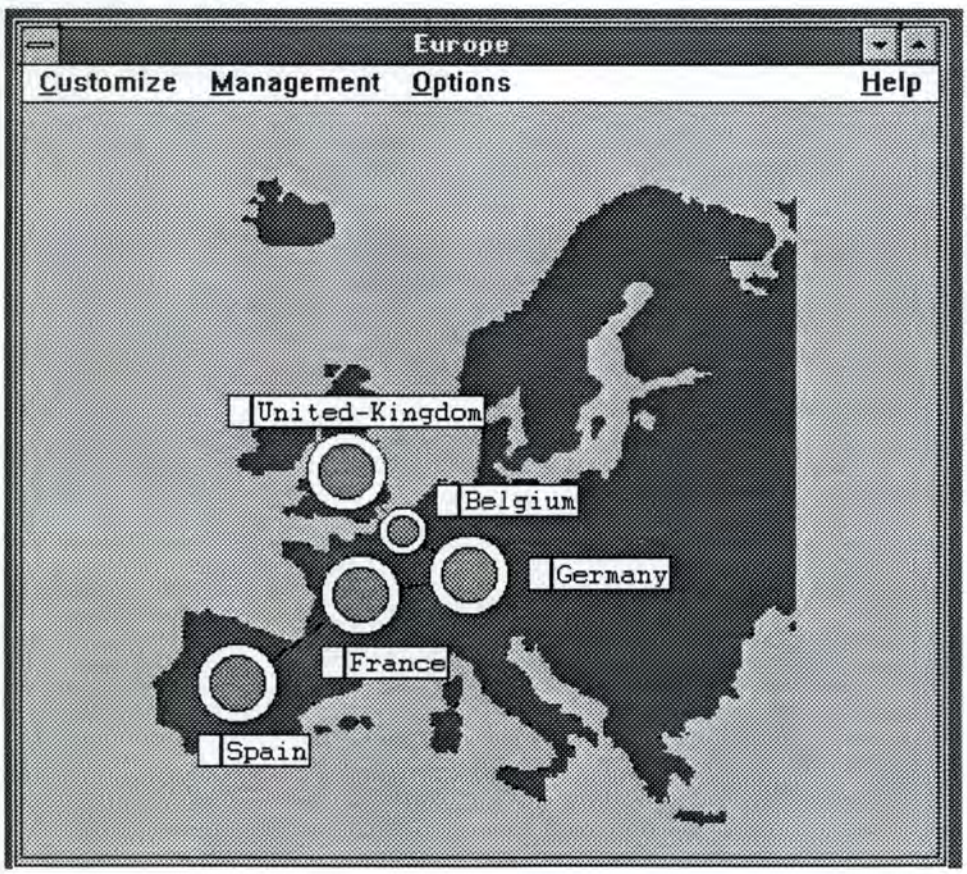


Figure 3.3 :Système de cartes du/des réseau(x).

Cette application permet à l'utilisateur de construire un modèle du réseau géré. Ce modèle doit en permanence être cohérent par rapport aux données se trouvant dans la base de données. Etant donné qu'il n'existe pas de fonctionnalité d'auto-découverte, c'est l'administrateur-système qui aura dû rentrer les données en spécifiant les éléments gérés et leurs interconnexions. Les éléments ainsi encodés devront respecter une certaine hiérarchie, c.-à-d. qu'un sous-réseau sera composé, par exemple, d'un ensemble de PC, d'un concentrateur et d'un routeur, le concentrateur étant lui-même composé de plusieurs portes.

Les éléments pourront être organisés par chaque utilisateur pour représenter une vue logique ou physique ou encore toute autre vue partielle du/des réseaux gérés (par exemple, une vue de tous les routeurs et de leur(s) interconnexion(s)) (Figure3.4). Afin de les rendre plus compréhensibles, il est possible d'ajouter une image de fond qui représentera une carte géographique ou un plan d'immeuble. A partir des instances représentées sur une vue, il est possible de déclencher la sélection des problèmes propres à chaque instance, ou encore d'accéder à un mémo qui lui est attaché. L'utilisateur pourra également créer, modifier et détruire les symboles d'une carte.

Outre ces vues, il est possible d'avoir une vue hiérarchique, au moyen d'un arbre des divers composants, et d'aller immédiatement appeler la "vue" correspondante d'un élément ou d'un sous-réseau.

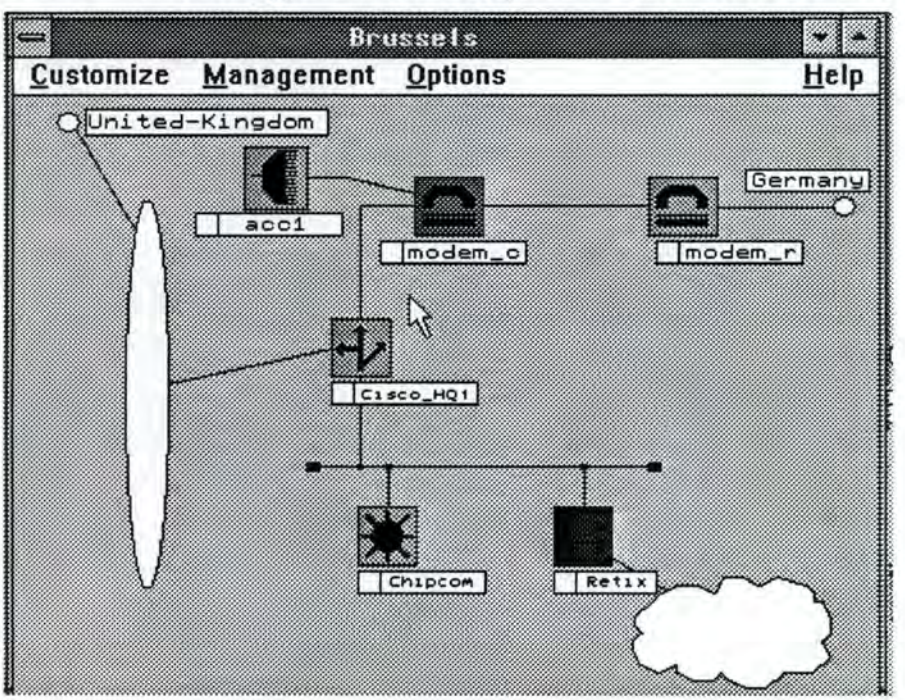


Figure 3.4. : Vue du sous-réseau.

- Le *gestionnaire de performances* permet d'accéder à un grand nombre d'applications et d'options de performances spécifiques au PSM. Il rend possible le contrôle des choses suivantes :
  - la spécification des caractéristiques d'interrogations et des valeurs limites pour le déclenchement d'alarmes,
  - la spécification du format des données de limites,
  - la récolte et la visualisation de statistiques historiques et en temps réel,
  - l'affichage d'un résumé d'interrogation,
  - l'affichage de données spécifiques à un élément.
- Le *gestionnaire des fautes* est par définition un des éléments clés puisque, par essence, le système de gestion devra permettre la prise de connaissance des informations relatives au contexte de la faute. NMC Vision considère trois types de fautes :
  1. une interrogation (polling) qui n'obtient pas de réponse dans le délai fixé,
  2. le passage d'une valeur limite fixée pour un attribut,

3. une information non sollicitée provenant d'un sous-système (trap).

Outre leur type, les fautes se verront également affectées d'un degré de gravité pouvant aller de 1 à 5.

Lorsqu'une faute est détectée, le symbole de la carte qui représente l'élément concerné clignote (ou un élément supérieur le contenant s'il n'est pas présent dans la vue active); une entrée active est créée dans la boîte de dialogue des fautes actives et la fenêtre de statut est mise à jour. Bien entendu, les renseignements au sujet du problème seront stockés dans la base de données et un trouble ticket sera créé. Pour voir les renseignements au sujet de la faute, on pourra soit consulter le trouble-ticket ou le rapport de faute, soit générer un rapport de faute(s) ou encore imprimer le contenu de l'enregistrement de la base de données.

- Le *sous-système de protocole* est certainement l'élément le plus complexe. En fait, cet élément est attaché à un protocole de gestion et seules les fonctions ICMP "ping" et LLC-1 "echo" sont fournies par le système de base. Pour que le système puisse fonctionner, il est nécessaire d'adjoindre un sous-système de protocole; il semble qu'un sous-système SNMP soit fourni en standard avec NMC Vision 4000. Il offre en effet des fonctionnalités de gestion SNMP (par exemple : manipulation des "trap" et gestion de MIB). Un sous-système de protocole est composé de quatre éléments :

  - le "pinger" générique et le gérant des "ping",
  - le gestionnaire de communications spécifique au protocole,
  - le système d'interrogation et le gestionnaire de statut propres au protocole,
  - la suite de protocoles.

  Voyons ces éléments d'un peu plus près. Tout d'abord il y a le *"pinger" générique et le gérant des "ping"* qui interrogent le réseau afin de s'assurer de la présence d'un élément. Le "ping" est en fait une requête de bas niveau émise relativement fréquemment à destination d'un élément dont on cherche à vérifier l'existence. Pour leur part, le *système d'interrogation et le gestionnaire de statut propres au protocole* sont utilisés pour la récolte d'informations auprès des éléments du réseau. Ils créent une trame de requête à destination de l'élément concerné et la passent au *gestionnaire des communications* qui la transmet. Il faut encore préciser que le système d'interrogation contient lui-même la suite de protocoles, ce qui permet de limiter l'encodage entre les processus.

  Le *système d'interrogation* accepte trois types de demandes: l'interrogation à des fins de statistiques historiques, l'interrogation pour observer le passage de limite et l'interrogation pour obtenir des statistiques en temps réel. Lorsqu'il s'agit d'informations utilisées en temps réel, il n'y a aucune conservation dans la base de données et la fréquence d'interrogation doit être élevée. Par contre, les autres demandes seront espacées et leurs résultats stockés dans la base de données des performances.

  Le *gestionnaire des communications* est propre à chaque protocole. Il communique directement avec les objets gérés, mais passe par le *Request Broker* pour communiquer avec les autres processus tels que le système d'interrogation ou les PSM. Il offre un service "garanti" dans la mesure où il renvoie toujours une réponse, c.-à-d. que s'il n'obtient pas de réponse dans le délai fixé (time-out), il transmet alors une réponse négative signalant, par exemple, le time-out.

- La *base de données* et ses *outils* ont déjà fait l'objet de commentaires dans le point 3.1.9. Nous reprendrons donc ici ce qui a été dit à cette occasion. La base de données de NMC Vision est organisée au moyen de RAIMA Corporation's Data Manager, plus précisément db_VISTA. Ce système se présente comme une base de données rapide. Elle comporte trois parties: la base de données d'administration, de configuration et de performances.

  - La partie administration contient l'information à propos des divers sous-systèmes et appareils définis pour le système.
  - La partie configuration décrit la structure et l'interconnectivité des éléments physiques, ainsi que leurs caractéristiques.
  - La partie performances contient l'historique des informations de performance des éléments repris dans la base de données de configuration.

  Les *outils de gestion* de base de données permettent de supprimer et de sélectionner certaines données historiques sur base d'un critère. L'importation et l'exportation de fichiers ASCII sont également prévues. Les informations de performances peuvent être importées dans des applications de tiers (tableur ou base de données SQL).

- Les *PSM* sont le dernier type de composants essentiels, mais ne font pourtant pas partie de la base. Il s'agit d'applications permettant la gestion des appareils ou familles d'appareils du réseau. Ces modules implémenteront les fonctions nécessaires à la gestion FCAPS, mais la couverture de ces cinq domaines de gestion n'est pas imposée. Ils doivent cependant permettre la représentation sur la carte d'un équipement physique du type qu'ils manipulent, c.-à-d. qu'ils devront supporter la création, la modification et la suppression d'un équipement physique. A coté de ces fonctionnalités, certains systèmes implémentent des fonctions de chargement et de déchargement qui autorisent respectivement le chargement automatique de la configuration du système physique dans la base de données de configuration et inversement. Si ces fonctions ne sont pas implémentées, le PSM affichera une boîte de dialogue en vue d'obtenir les informations auprès de l'opérateur. Beaucoup de PSM offrent une représentation statique ou même interactive de l'équipement géré (Figure 3.5). De tels modules existent également pour la gestion des lignes et des sous-réseaux; ce sont respectivement les LSM (Line Specific Module) et les NSM (Network Specific Module).

  En option, il est possible de disposer d'un kit de développement de PSM qui offrira les API nécessaires, des exemples de codes en C++, des moyens d'aides, etc ... .

***Figure 3.5.** : Représentation dynamique d'un composant pour lequel existe un PSM.*

3.2.3. Un exemple de fonctionnement.

Partons d'un système configuré, c.-à-d. un système tel que la base de données d'administration et de configuration contient toutes les informations relatives aux équipements gérés. L'opérateur décide de placer une limite sur la valeur d'un attribut, soit pour fixer les idées, le nombre de paquets erronés. Suivons le cheminement des opérations; pour ce faire nous allons travailler en deux colonnes. La colonne de gauche reprendra les actions de l'utilisateur, tandis que celle de droite contiendra les activités des processus du système.

Opérateur	NMC Vision
Sélection de l'équipement visé.	L'interface graphique envoie le message au Request Broker (RB) qui le transmet au PSM qui se signale auprès du RB comme étant actif.
Sélection de la fonction d'établissement d'une limite.	Le PSM fait appel au gestionnaire de performances. La réponse de celui-ci permet l'affichage de la boîte de dialogue adéquate.
Envoi des paramètres.	Le PSM envoie un message contenant les paramètres au RB qui les sauvegarde dans la base de données en les transmettant aux outils de la base de données; il les transmet également au gestionnaire de performances. Le *système d'interrogation et le gestionnaire de statut propres au protocole* sont utilisés pour la récolte d'informations auprès des éléments du réseau. Ils utilisent les informations émanant du *gestionnaire de performances* qui lui parviennent par l'intermédiaire du *Request Broker*. Après avoir créé une trame de requête à destination de l'élément concerné, il la passe au *gestionnaire des communications* qui la transmet. Les réponses sont analysées par le gestionnaire de

performances et stockées dans la base de données des performances.

Supposons que la limite soit dépassée.

Le gestionnaire de performances le signale au RB qui transmet le message au gestionnaire des fautes qui lui-même déclenche le signal d'alarme sur la carte par l'intermédiaire d'un message envoyé au *système de cartes*; une entrée active est créée dans la boîte de dialogue des fautes actives et la fenêtre de statut est mise à jour. Un trouble ticket sera créé. Eventuellement, il y a déclenchement automatique d'un scénario de réduction de faute.

Plusieurs possibilités s'offrent à l'opérateur.

1. Parcours hiérarchique des cartes pour en arriver à l'élément et sélection des fautes qui lui sont propres par l'intermédiaire d'un double click sur le symbole adéquat.

1. Le *système de carte* parcourt le modèle stocké dans la base de données pour afficher la hiérarchie des éléments. Le double-click déclenche une sélection des alarmes relatives à cette instance.

2. Consultation de la boîte de dialogue des fautes actives (Figure3.6) et ouverture du trouble ticket et/ou zoom, c.-à-d. ouverture directe de la vue de l'élément fautif.

2. Le système de fautes envoie un message demandant la recherche des informations relatives à cette erreur et/ou active le PSM du type d'élément fautif pour afficher la vue du problème.

3. Génération d'un rapport de faute et/ou impression.

3. Génération d'un rapport par le gestionnaire de fautes en requérant les éléments dans la base de données.

Exécution d'une de ces actions.

Prise en charge de la faute.

Changement de statut: l'alarme devient active-acknowledged (active et prise en charge)

Réduction de la faute en modifiant la configuration d'un appareil sélectionné.

Activation du PSM correspondant à l'appareil dont on veut modifier la configuration et appel de la fonction de configuration qui ira rechercher les données de la base de données de configuration. Après modification, la cohérence est vérifiée et la configuration modifiée est sauvegardée en mémoire. Le trouble ticket est mis à jour et l'alarme est fermée.

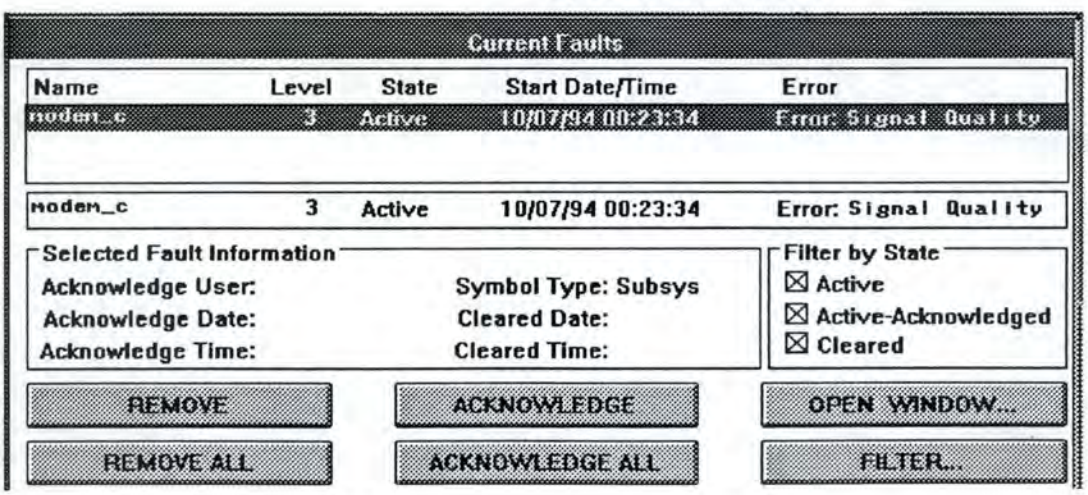


Figure 3.6. : Boîte de dialogue des fautes actives.

3.3. Etude détaillée de ALCATEL - NM-Expert.

Nous avons choisi de détailler ce produit pour plusieurs raisons. La première, c'est sa similitude avec le modèle présenté au chapitre 2. Ensuite, c'est le fait qu'il se fonde sur un système expert pour assurer la gestion et aider l'opérateur. Enfin la troisième, c'est qu'il s'agit d'un système capable de fonctionner comme un gestionnaire de gestionnaires qui peut également jouer le rôle de plate-forme pour intégrer diverses applications de gestion.

3.3.1. Architecture détaillée : processus de raisonnement (PR).

NM-Expert est un processus de raisonnement avec des interfaces vers le monde extérieur (Figure 3.7.). Le processus de raisonnement se compose d'une base de connaissances organisées suivant un certain formalisme et d'un moteur d'inférence. Le formalisme utilisé pour créer les connaissances fait partie du système NM-Expert. Nous allons ici nous pencher sur le coeur du système, ensuite nous étudierons ses relations avec l'extérieur.

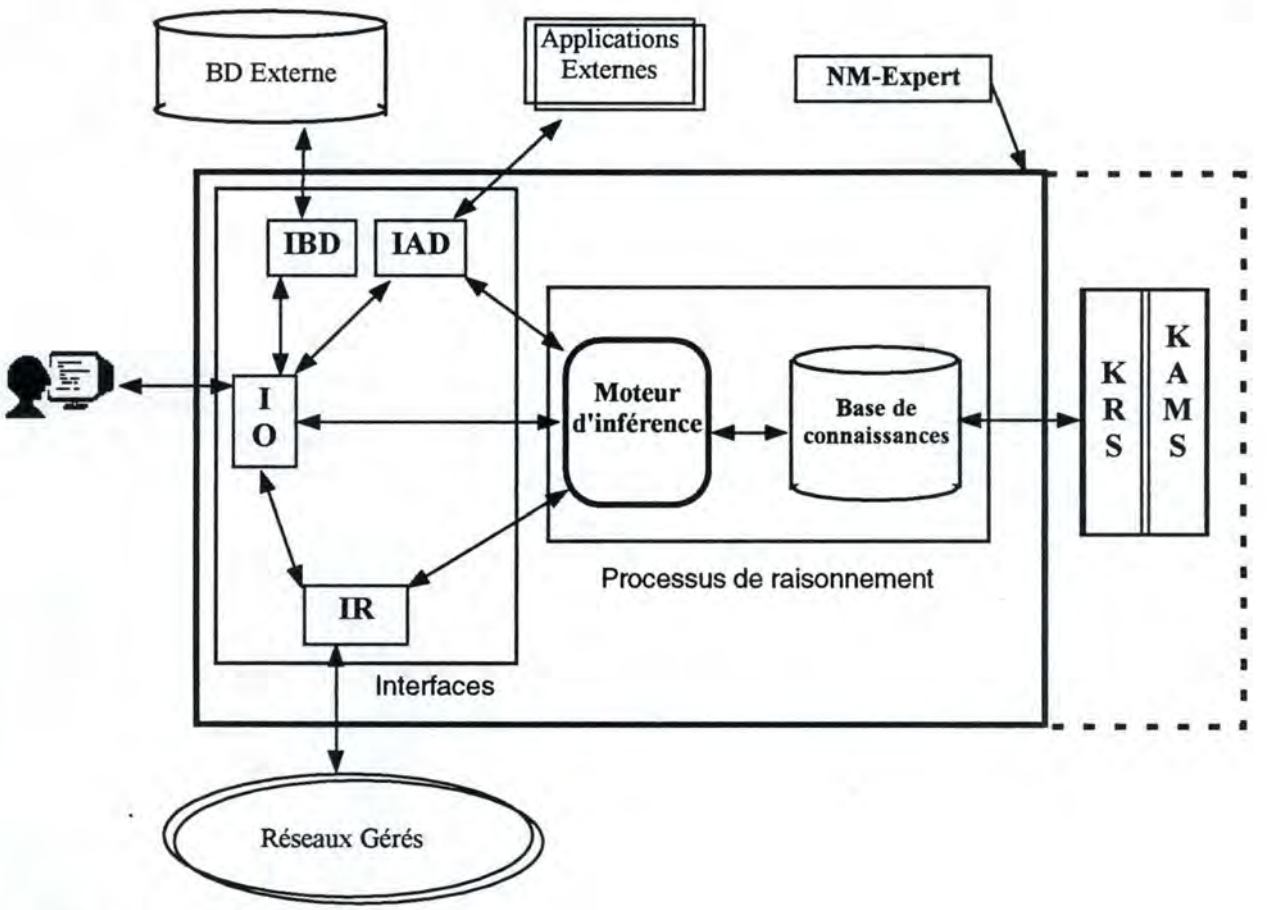


***Figure 3.7.** : Architecture de NM-Expert.*

- La ***base de connaissances***. Les connaissances utilisées par NM-Expert sont de deux ordres : les connaissances structurelles et les connaissances heuristiques.

  Les *connaissances structurelles* se décomposent à leur tour en connaissances *statiques* qui décrivent le(s) réseau(x) géré(s) ainsi que les évènements pouvant être émis par un opérateur ou par les composants, et en connaissances *dynamiques* qui sont l'ensemble des déductions à prendre en considération lors du raisonnement. Dans la partie dynamique, on retrouve toutes les informations à court et moyen terme qui évoluent continuellement. Ce sera, entre autres, le cas des renseignements relatifs au statut des divers éléments gérés ainsi que les réponses issues de requêtes et momentanément stockées. Si l'on désire conserver des renseignements à plus long terme, on aura alors recours à une base de données extérieure au système.

  Les connaissances heuristiques expriment les règles à appliquer lorsqu'on raisonne au sujet du réseau géré ou des évènements qu'il émet. Ces règles permettront de détecter, diagnostiquer et résoudre les problèmes rencontrés.

Afin d'être cohérentes et de pouvoir être exploitées par le moteur d'inférence, ces connaissances doivent respecter un certain formalisme. Les règles de celui-ci constituent le Knowledge Representation System (KRS - Système de représentation des connaissances). Pour chaque type de connaissances, il existe un formalisme qui décrit toutes les caractéristiques et propriétés du type de connaissances concerné. Pour illustrer ces explications, nous utiliserons les deux exemples trouvés dans la brochure de NM-Expert, c.-à-d. les formalismes des descriptions de connaissances structurelles au sujet des éléments du réseau et au sujet des évènements.

Les connaissances au sujet du réseau font l'objet de trois types de vues.

- Tout d'abord, il y le *modèle du réseau*. Proche du modèle ERA (Entities Relationship Attributes), il permet de caractériser le type de réseau. Un réseau X.25 n'a pas les mêmes composants qu'un LAN; il s'agit donc de décrire tous les types de composants et les relations entre ceux-ci. Ainsi, si un réseau contient des noeuds où sont branchées une carte réseau, une carte-mère et une carte mémoire, alors le modèle définira les types d'éléments suivants : noeud, carte réseau, carte mère et carte mémoire. En plus, le modèle devra définir la relation "branché dans" qui liera un type de carte à un noeud.

- Ensuite, il y a la *configuration du réseau* qui est en fait l'instanciation du modèle. On décrira les instances des types de composants qui existent dans le réseau. On aura par exemple trois noeuds: un à Liège, un à Bruxelles et un à Anvers; sur chacun seront branchées un certain nombre de cartes des types décrits ci-avant qui seront également instanciées.

- Enfin, il y aura les *données descriptives* qui sont des renseignements utiles mais pas utilisés dans le processus de raisonnement; par exemple, le numéro de téléphone du responsable local d'un noeud.

La connaissance au sujet des évènements est également spécifique à un type de réseau ou à son protocole. Afin de pouvoir prendre le message en considération dans le processus de raisonnement, il faut pouvoir décrypter les informations qu'il contient. Grâce à ces connaissances, NM-Expert va pouvoir transformer le message en un objet-évènement, un évènement générique. Le mécanisme de transformation est généré automatiquement par le système; il est une partie de la passerelle de gestion que nous avions décrite au chapitre 2.

Les connaissances heuristiques sont des règles qui permettent de raisonner sur les évènements, les problèmes et l'état du réseau. La définition d'une règle comporte plusieurs parties: la définition de l'objet auquel s'applique la règle, la condition de déclenchement et la règle en elle-même. Cette règle est de la forme classique IF ... THEN ... ELSE ... . Elle pourra déclencher une/des action(s) et/ou un raisonnement, et/ou générer une/des déduction(s). Ces dernières viendront compléter la base de connaissances. Une règle peut être déclenchée par :

- un évènement (provenant de l'opérateur ou du réseau),
- une réponse du réseau à une commande,
- une déduction,
- un "time out" (limite de temps),
- une question posée,
- un résultat d'une requête adressée à une base de données.

Les raisonnements déclenchés porteront entre autres sur :

- des analyses de situations et des corrélations d'évènements,
- la détection de problèmes et leur diagnostic,
- la corrélation avec des problèmes déjà signalés.

Pour la saisie de ces connaissances, il existe un système d'acquisition et de maintenance des connaissances (Knowledge Acquisition and Maintenance System). Il offre un "browser" permettant l'affichage de la hiérarchie de classes, un éditeur de classe ainsi que divers outils de gestion. Son utilisation facilite la création et la modification des connaissances.

A coté des connaissances qui doivent être créées par l'administrateur, il en existe certaines qui sont prédéfinies. Il est important de comprendre que la création d'un système de gestion à partir de NM-Expert, c'est surtout la production des données de la base de connaissances.

- Le ***moteur d'inférence***. Il a été conçu de manière à optimiser l'inférence dans le cadre de la gestion de réseau(x). Il permet le raisonnement en temps réel et peut être déclenché, comme la plupart des systèmes non experts, par des évènements externes (provenant du réseau ou de l'opérateur); en outre, il réagira aux évènements internes tels que les déductions produites par un raisonnement. Il est aussi capable de raisonner sur des hypothèses et sur la chronologie des évènements. Enfin, il est capable de traiter plusieurs problèmes simultanément, ce qui peut être très important dans le domaine des télécommunications où une action se propage souvent et où l'on peut être en présence d'un nombre important de problèmes.

  Le moteur d'inférence est en fait l'élément actif du processus de raisonnement. C'est lui qui va déclencher les actions contenues dans les règles et qui va "raisonner". En fait, il est capable de faire trois types de raisonnements: le raisonnement au moyen de règles, le raisonnement sur le modèle du réseau et le raisonnement sur le temps.

  Il peut tout d'abord raisonner en se basant sur les règles; dans ce cas, il utilisera des fonctions de manipulation des déductions, c.-à-d. de destruction, de confirmation, de création, de copie et de recherche de déductions. Un autre mécanisme puissant et essentiel pour ce raisonnement est celui qui veille à la cohérence des déductions: lors de la destruction d'une déduction, il veillera à mettre à jour toutes les déductions qui se sont fondées sur elle. Cela est aisément réalisable grâce à la double organisation des déductions.

  Toute déduction est toujours associée à un objet du réseau géré; elle appartient à son histoire. Il est donc aisé de retrouver toutes les déductions actives relatives à un objet. A coté de cette organisation, chaque déduction fait également partie d'un "réseau" reliant toutes les déductions qui caractérisent une situation et qui y sont relatives.

  La deuxième forme de raisonnement se base sur le modèle et la configuration du réseau stockés dans la base de connaissances. Ainsi, en se basant sur les relations entre les types d'éléments auxquels appartiennent les objets, il est possible de faire des déductions. Par exemple, si un noeud est "down", tous les périphériques y connectés deviennent inaccessibles.

  La dernière forme de raisonnement se base sur le temps, c.-à-d. sur la chronologie des évènements ou encore vis-à-vis de "time out". Cela permet de déclencher des actions à certains moments fixés dans le temps ou d'analyser les données de temps associées à des données ou des déductions.

3.3.2. Architecture détaillée : les interfaces.

Quatre types d'interfaces existent dans NM-Expert (Figure 3.7.); elles ont pour but de permettre au processus de raisonnement d'interagir avec le monde extérieur, c.-à-d. les réseaux gérés, l'opérateur, les bases de données et d'autres applications. Au point 3.3.3 nous verrons comment les informations circulent dans NM-Expert; nous étudierons ici uniquement les interfaces, leurs fonctions et leur constitution.

- L'***interface opérateur (IO)***. De type graphique, elle se base sur les standards X Window et OSF/Motif. Elle peut faire l'objet de nombreuses personnalisations dont, entre autres, l'ajout de fonctions et de fenêtres, la personnalisation des représentations graphiques, la modification de ses paramètres (par exemple : la langue employée) et enfin l'adaptation automatique aux données de la base de connaissances (par exemple : attributs d'un nouvel objet).

  On s'en doute, c'est grâce à cette interface que l'opérateur pourra remplir sa tâche. Elle disposera donc de plusieurs fonctions utilisées tantôt pour l'affichage, tantôt pour la saisie de données ou de commandes.

  - La gestion de problèmes: deux fenêtres permettent d'effectuer les opérations nécessaires à la résolution des problèmes ainsi qu'à leur suivi. Il y a la fenêtre des problèmes qui autorise diverses opérations telles que la description et l'explication des problèmes. Il y a également une fenêtre qui permet de suivre le déroulement du scénario et par laquelle se fera l'interaction entre l'opérateur et le scénario de résolution de problème.

  - La visualisation graphique du réseau: la représentation ainsi offerte permet de consulter les renseignements à propos des objets gérés et autorise les zooms avant et arrière. Cela permettra encore de visualiser tout problème par un changement de couleur.

- La gestion de la configuration: l'opérateur peut visualiser, modifier, ajouter ou détruire tout objet du réseau au moyen de la fenêtre de configuration.
- L'interaction avec les éléments du système: l'opérateur peut par ce moyen envoyer des commandes aux éléments du réseau, faire des requêtes dans une base de données ou répondre à des questions posées par le moteur d'inférence.
- La notification: le système peut envoyer des messages, des notifications à l'opérateur, par exemple, suite à un évènement pour lequel la règle appliquée contient cette action.
- Le contrôle d'accès aux données et fonctions: un double contrôle d'accès existe pour accéder aux données et aux fonctions de NM-Expert. Le premier est le mécanisme de login de UNIX, tandis que le second est un contrôle du profil. NM-Expert connaît trois types de profils: développeur, gestionnaire et opérateur. A chaque profil correspond un certain nombre de fonctions autorisées.
- La simulation et l'affichage de tous les évènements: ces deux fonctions ne sont utiles que pour les développeurs qui peuvent ainsi utiliser l'ensemble des informations pour construire et suivre le déroulement de leurs applications.

Signalons encore que cette interface est multi-opérateurs et permet donc à plusieurs opérateurs de gérer simultanément le même système.

- L'***interface réseau (IR)***. C'est en fait la passerelle de gestion définie au chapitre 2. C'est par celle-ci que passent toutes les communications avec le(s) réseau(x). Elle est composée de deux couches.

  La couche supérieure est composée, entre autres, du mécanisme de transformation automatiquement créé au moyen des descriptions des évènements et commandes contenues dans la base de connaissances. Il transforme les messages en objets manipulés par NM-Expert et inversement. Cette couche cache donc les détails des formats natifs utilisés dans les protocoles des réseaux.

  La couche inférieure est composée d'une multitude de modules (DCP - Data Communication Process) qui permettent chacun la communication de NM-Expert suivant un protocole avec les éléments du/des réseau(x).

  La communication entre les deux couches se fait au moyen de connexions logiques (une par DCP).

- L'***interface d'application distante (IAD)***. Elle est basée sur le même principe que l'interface de réseau. Elle permet d'accéder aux fonctionnalités offertes par des applications externes au système et développées par des constructeurs ou des utilisateurs. Encore une fois, une commande logique est émise à destination de l'interface où la traduction s'opère et la commande peut alors être lancée. Par exemple, elle pourrait donner accès à une application de gestion de la comptabilité.

- L'***interface de base de données (IBD)***. Comme nous l'avons déjà dit, NM-Expert ne dispose d'aucune base de données; il n'utilise que sa base de connaissances. S'il est nécessaire d'accéder à des informations stockées dans une base de données, il faudra alors passer par une interface; ce sera entre autres le cas lorsque le moteur d'inférence ou l'opérateur désireront accéder à des informations historiques. Cette interface permet à NM-Expert de ne pas être lié avec un quelconque système de gestion de base de données (SGBD).

  Le principe est toujours le même; il faut passer par des requêtes logiques traduites dans l'interface pour être transmises au SGBD sous une forme qu'il comprend. Nous voyons ici l'intérêt de passer par des requêtes logiques. En effet, si l'on doit accéder à plusieurs bases de données utilisant différents SGBD, les commandes restent les mêmes puisque chaque SGBD a son propre module d'accès à la base de données qui fera la traduction (DBAM - Data Base Access Module). Ce DBAM est bien entendu construit à l'aide de la base de connaissances.

3.3.3. Flux de données.

Nous allons ici décrire le type de données qui circulent entre les divers composants de NM-Expert (Figure 3.7.).

- IR ==> PR : deux types de messages circulent entre ces éléments, ce sont:
  - des évènements (informations non sollicitées) issus des composants des réseaux gérés,

- des réponses aux commandes lancées par le processus de raisonnement lui-même ou par l'opérateur au moyen de son interface et dont il a demandé que la réponse soit transmise au processus de raisonnement.

- PR ==> IR : seules des commandes-réseau logiques sont envoyées par le processus de raisonnement.
- RP ==> IO : quatre types de données peuvent être émis :
  - des problèmes du/des réseau(x) signalés à l'opérateur afin qu'il les examine,
  - des questions posées par le processus de raisonnement pour obtenir des informations supplémentaires,
  - des notifications, c.-à-d. des messages générés par NM-Expert (messages du système),
  - des commandes-réseau logiques suggérées à l'opérateur ou pour lesquelles le processus de raisonnement demande à l'opérateur l'autorisation de lancement.
- IO ==> RP : ici encore plusieurs types de messages sont possibles:
  - des évènements de l'opérateur qui sont en fait des saisies manuelles,
  - des réponses aux questions posées par le processus de raisonnement,
  - des changements de configuration, c.-à-d. que l'opérateur peut ajouter, retirer ou modifier des éléments du/des réseau(x) et également altérer les relations avec d'autres objets.
- IO ou PR ==> DBI ou IAD : il s'agira, on s'en doute, d'envoyer des commandes ou des requêtes logiques.
- DBI ou IAD ==> IO ou PR : ce seront fort logiquement les réponses aux requêtes et commandes précédemment envoyées.

3.3.4. Comparaison avec le modèle du chapitre deux.

L'*interface utilisateur* reprend les fonctions de *l'interface utilisateur unifiée* et du *module de présentation des informations de gestion de réseau(x).*

Le *processus de raisonnement* regroupe une partie des *applications de gestion intégrée* et des *éléments d'application*. Les autres applications et leurs éléments sont repris par les *applications distantes* intégrées grâce à *l'interface d'application distante.*

La *base de connaissances* reprend une partie de la *MIB*; le reste est situé dans les composants eux-mêmes.

L'interface réseau reprend les fonctions de *passerelle de gestion* et intègre les divers *protocoles.*

3.4. Un système de trouble ticket (rapport de problème) - REMEDY Action Request System.

Dans les points précédents, nous avons parlé de systèmes de trouble ticket à de nombreuses reprises. Pour rappel, un trouble ticket est l'enregistrement électronique d'un problème rencontré. Divers renseignements tels que l'heure à laquelle s'est produit le problème et l'état du composant en cause à ce moment, seront stockés dans une base de données; on y ajoutera également la solution apportée afin de faciliter le diagnostic et la résolution de problèmes semblables ultérieurement. REMEDY Action Request System (ARS) est certainement le système de trouble ticket le plus populaire. Afin d'éclairer le lecteur, nous allons ici en dire quelques mots.

ARS est une application client/serveur distribuée. Le serveur gère la base de données reprenant les problèmes soumis ainsi que ceux qui ont été résolus. Il tourne sur une/des station(s) UNIX et supporte un nombre illimité de clients. La partie client tourne sur des stations de travail et/ou des PC et fournit l'interface utilisateur qui peut être de type OPEN LOOK, OSF/Motif ou Windows; cela offre donc une grande flexibilité. Le serveur communique avec les clients au moyen de RPC (Remote Procedure Call - Appel de procédure à distance). Les fichiers sont soit des fichiers UNIX (Flat File), soit des bases de données SQL. Le serveur offre en plus une API permettant une totale personnalisation et une intégration avec d'autres outils, des systèmes de messagerie électronique, le téléphone ou encore un système de gestion de réseau(x) (intégré ou non).

L'émission d'un trouble ticket peut être déclenchée de plusieurs façons. Nous retenons les suivantes: un utilisateur confronté à un problème envoie un trouble ticket à partir de son poste ou à partir d'un poste voisin. Il peut encore téléphoner au bureau d'aide (Help Desk) ou au centre de contrôle où son interlocuteur remplira un

trouble ticket afin de signaler le problème. Enfin, suite à un problème de communication, un évènement déclenche une alarme et un trouble ticket est automatiquement rédigé par le système de gestion de réseau(x) sur base des éléments repris dans l'alarme.

La présentation des trouble tickets est entièrement personnalisable par l'administrateur-système. Plusieurs modèles seront conçus afin de correspondre au recueil de toutes les informations nécessaires à la résolution du problème; de plus, une aide est disponible pour l'utilisateur soumettant un trouble ticket. Notons encore qu'il est possible de prévoir des flux distincts pour chaque type de trouble ticket.

Pour aider à la résolution, de puissants mécanismes de recherche existent. Il est ainsi permis de rechercher parmi les trouble tickets préexistants ceux ayant des symptômes similaires; la recherche ne se limitera pas aux données de l'entreprise, mais pourra être étendue à l'expérience de clients ou de vendeurs.

Terminons en disant qu'il est possible de faire des rapports et des analyses se basant sur les données contenues dans la base de données.

3.5. Un système intelligent de gestion d'évènements. TIVOLI - TEC (Tivoli/Enterprise Console).

Nous avons pu nous apercevoir que la base de la gestion des fautes et anomalies est la réception et l'analyse des évènements émis par les composants des réseaux gérés. Nous nous proposons de voir ici un système intelligent de gestion des évènements: il s'agit de Tivoli/Enterprise Console (TEC). TEC est une application de l'environnement de gestion de Tivoli (Tivoli Management Environnement). Elle permet la gestion des évènements en se basant sur des règles (rules-based) et elle intègre la gestion de réseau(x), de systèmes de base de données et d'applications; elle récolte, traite et répond aux évènements de gestion. TEC centralise les alarmes et les évènements issus de diverses sources.

Avant toute chose, nous expliciterons les deux concepts clés de ce système : les *évènements* et les *règles*.

Un *évènement*, c'est tout changement survenant dans l'état d'une ressource système ou dans une application; il peut indiquer différentes choses. Chaque évènement émane nécessairement d'une source d'un type déterminé; virtuellement, TEC accepte tous les types *d'évènement* (par exemple : de réseau, de système, de performance, de base de données, d'application).

Les *règles* sont en fait les moyens permettant la corrélation *d'évènements* rendant possible la gestion efficace et efficiente des ressources informatiques de l'entreprise. Ces *règles* sont ici écrites en Prolog.

Trois types de composants sont à la base du fonctionnement de TEC: ce sont les adaptateurs d'évènement (Event Adapters), le serveur d'évènement (Event server) et les consoles d'évènement (Event consoles) (Figure 3.8).

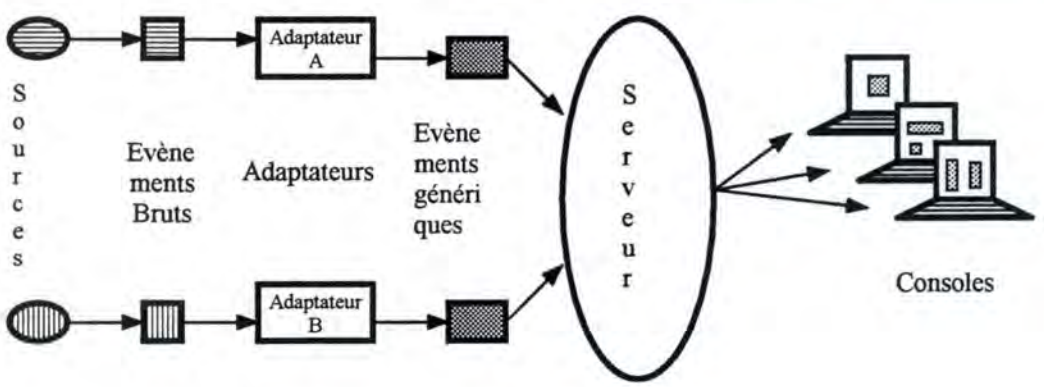


Figure 3.8.: Architecture de TEC.

Les adaptateurs sont distribués et de divers types; ils sont attachés à des sources qui leur sont propres et qui sont elles-mêmes d'un type particulier (HP OpenView, IBM NetView, ...). Leur rôle est d'automatiquement rapporter au serveur les évènements générés par les sources en les transformant en évènements répondant à un format déterminé et générique (Figure 3.3). Remarquons qu'il est possible de configurer les adaptateurs afin qu'ils n'émettent pas les évènements inutiles; ceci permet une réduction du trafic sur le réseau.

Le serveur est central et remplit plusieurs fonctions; c'est l'élément principal de cette application. Parmi ces fonctions, nous en retiendrons quelques-unes qui nous semblent être particulièrement intéressantes.

- Logging : tous les évènements sont validés et stockés dans une base de données. Un tampon permet la temporisation du traitement de ces évènements.
- Passage par les règles : après création et/ou modification, les évènements sont passés en revue afin de s'assurer de l'existence d'une règle applicable. Si tel est le cas, la règle est appliquée, c.-à-d. qu'on exécutera les actions prévues par celle-ci. Par exemple, on pourra:
    - modifier les attributs de l'évènement et/ou d'autres évènements,
    - éviter l'affichage d'autres évènements,
    - exécuter une application ou un script,
    - réévaluer l'ensemble des évènements,
    - écarter l'évènement.
- Corrélation des évènements: les règles peuvent réduire le nombre de réponses requises par le système. Ainsi, lorsqu'on sait qu'un évènement A cause l'apparition de l'évènement B, on pourra fermer B dès son apparition si A est présent.
- Réponse automatique : une réponse automatique peut être déclenchée suite à un évènement.
- Adaptation des affichages des consoles : si un évènement a été fermé, il s'agit de le fermer partout afin d'éviter sa prise en charge par un autre opérateur.
- Escalation : s'il n'y a pas de réponse d'un opérateur dans un certain délai, une action est automatiquement exécutée par le serveur en vue d'augmenter le degré de gravité inhérent au problème.

Les consoles sont distribuées; elles permettent aux opérateurs de visualiser et de répondre aux évènements. Deux fenêtres sont affichées; la première permet la surveillance des sources, tandis que la seconde facilite la surveillance de groupes. Un *groupe d'évènements* est une configuration sur le serveur dans laquelle sont regroupés des évènements relativement semblables représentés par une alarme unique (par exemple : le groupe des imprimantes). On définira également le statut des divers utilisateurs au moyen des quatre profils disponibles; ceci rendra possible les réponses aux évènements marqués d'un niveau de prise en charge.

Chapitre 4 : Le système idéal ou le cas " M.B.B.".

Dans ce chapitre nous allons tenter d'illustrer tous les propos que nous avons tenus jusqu'ici en nous basant sur le cas de la société `M.B.B.` . Celle-ci est une société fictive (`M.B.B.` = Mémoire Benoit Boulanger) dont nous avons imaginé tant l'infrastructure que les activités. Nous avons voulu choisir une société possédant une infrastructure classique répartie sur différents sites dans le pays. Ce chapitre sera organisé suivant la chronologie des opérations de mise en place d'un SIGR. Nous commencerons en présentant l'infrastructure informatique préexistante. Ensuite, nous étudierons le projet d'implantation dans son ensemble. Pour terminer, nous étudierons les critères à prendre en considération. Il s'agira, en quelques sortes, d'établir un cahier des charges techniques. Les deux dernières étapes, le choix et l'implantation, seront alors traitées assez rapidement et d'une manière générique. Dans notre cas, nous ne désirons pas émettre un avis sur la qualité de l'un ou l'autre produit; ce choix se justifie car nous n'avons pu procéder à des tests et nos informations sur les différents systèmes abordés au chapitre précédent ne sont qualitativement et quantitativement pas équivalentes.

4.1. Situation de `M.B.B.` .

4.1.1. Description de `M.B.B.`

M.B.B. est une entreprise de service (secteur tertiaire), par exemple une société d'assurances, dont l'organisation est décrite à la figure 4.1. Le siège central est situé à Bruxelles; on retrouve dans chaque province une succursale dont dépendent les courtiers.

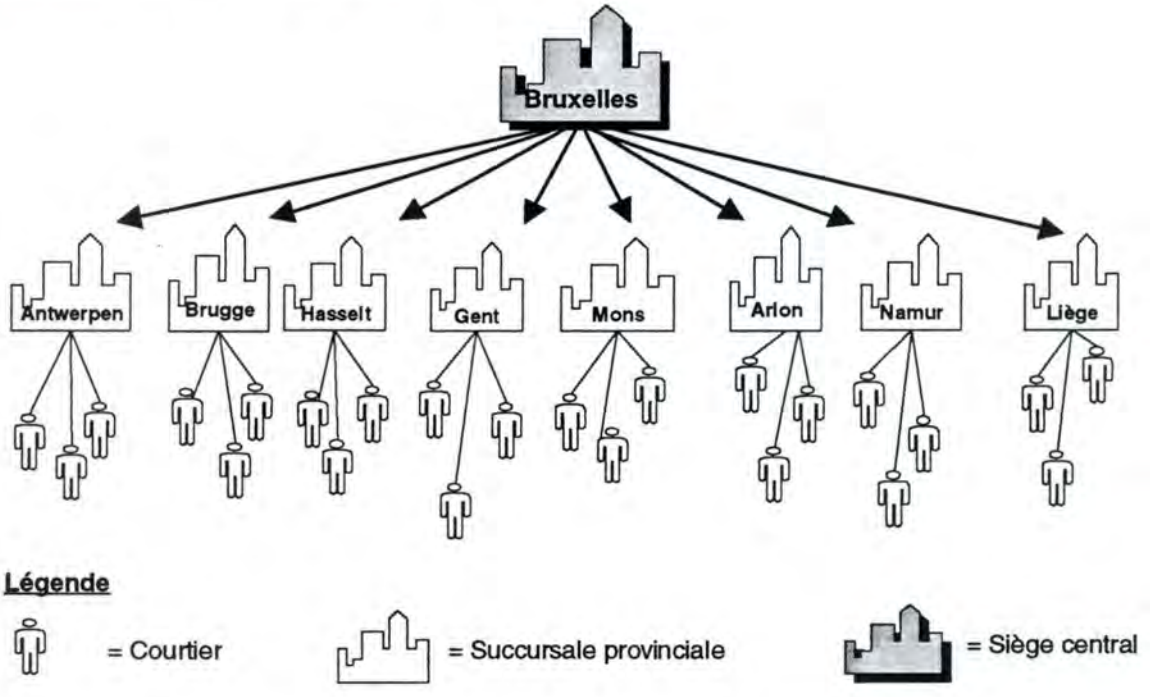


***Figure 4.1.** : Organisation de M.B.B. .*

4.1.2. Infrastructure informatique et d'appui à l'informatique (Figure 4.2).

Les bases de données clients, accidents, ... se trouvent au siège central. Ces bases de données sont compatibles SQL et actuellement gérées au moyen du SGBD (imaginaire) EGG. Celui-ci, ainsi que toutes les applications d'interrogations, de statistiques, ... tourne sur un système UNIX (imaginaire) MIC. Un second mainframe sert de back-up. Les salles des ordinateurs des sites provinciaux et nationaux sont équipées d'un système de climatisation.

Chaque succursale provinciale est reliée au système central au moyen d'une ligne louée et d'un hôte servant de routeur et de serveur de terminaux. Sur celui-ci tourne également une application veillant à la sécurité des accès. Chaque succursale dispose également d'un LAN fonctionnant sous Novell NETWARE 4.0.1. Celui-ci est connecté au routeur afin de permettre l'accès aux données du siège central. Outre le LAN, plusieurs modems

sont reliés à un multiplexeur et permettent aux courtiers d'accéder aux données du siège central soit depuis chez les clients via un portable et un modem, soit à partir de leur bureau. Ce multiplexeur est connecté à l'hôte provincial.

Enfin, chaque succursale est également équipée d'un PABX *. De plus, les équipements de chaque siège (PABX, serveur, routeur et concentrateur du LAN) sont rassemblés dans la salle des ordinateurs.

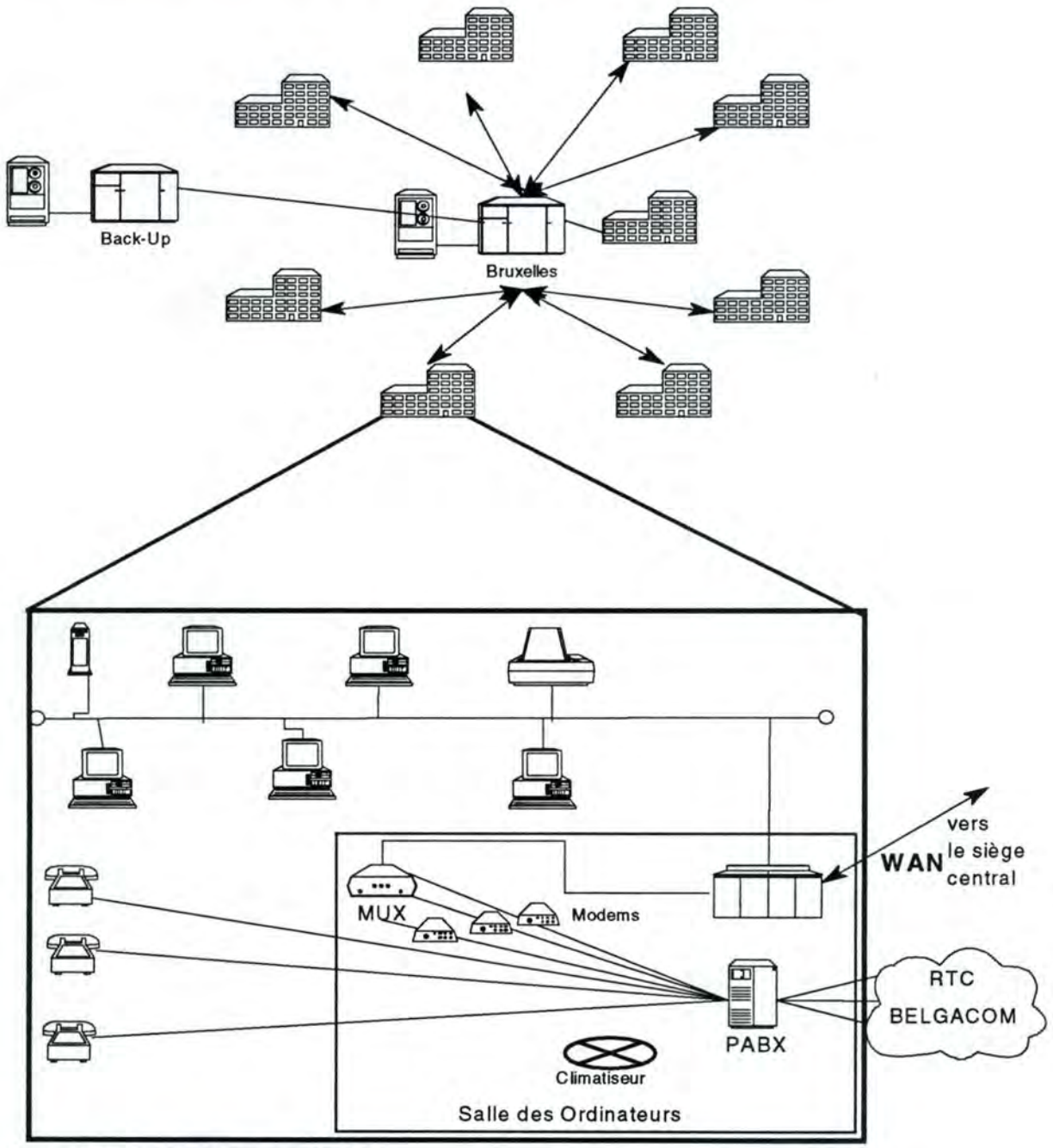


Figure 4.2. : Infrastructure informatique de M.B.B. .

4.1.3. Infrastructure de gestion existante.

Lorsque M.B.B. désire acquérir un SIGR, la situation en matière de gestion des systèmes est la suivante:

* Private Automatic Branch eXchange = Central téléphonique privé connecté à plusieurs lignes vers l'extérieur et auquel sont connectés les postes internes. Outre l'accès vers l'extérieur, il offre des services propres au groupe de postes qui y sont connectés (par exemple : conférence, signal d'appel, ...) .

- Le WAN est doté d'un système de gestion développé par le centre de traitement de l'information de M.B.B (CTI) fonctionnant sur base de commandes propres transmises à l'/aux hôte(s) distant(s) sous forme de chaînes ASCII (fonctionne suivant RPC*) (Figure 4.3a.). Ce système de gestion permet de contrôler le système central et les hôtes répartis. Le WAN est donc administré à partir de Bruxelles.
- Les LAN sont administrés localement à partir des "outils" fournis avec Novell NETWARE 4.01 (Figure 4.3b.). Novell NETWARE 4.01 supporte SNMP. La plupart des éléments du réseau local supportent également SNMP.
- Les PABX ne permettent que l'interrogation de leur base de données locale ainsi qu'une observation en temps réel de leur état (Figure 4.3c.). Aucune commande ne peut leur être adressée. Ils peuvent être pourvus d'une carte les rendant compatibles avec SNMP.
- Le système de climatisation est muni de divers capteurs et permet l'envoi de messages ASCII au moyen d'une ligne série asynchrone (Figure 4.3d.). De plus, il accepte des commandes élémentaires également envoyées sous forme ASCII et respectant un protocole propre, ceci moyennant une connexion au système destinataire via une simple paire torsadée.
- Les modems et le multiplexeur sont gérés au moyen d'un système développé par leur constructeur (Figure 4.3e.). Toutefois, le gestionnaire de ceux-ci peut travailler comme agent SNMP (proxy). Il est évident qu'il est alors subordonné à une station de gestion.

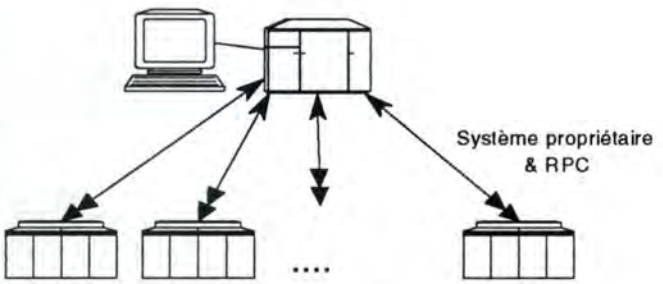


Figure 4.3a : Gestion du WAN.

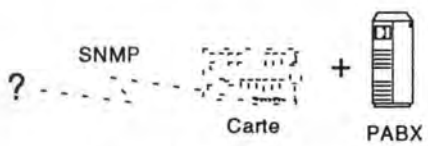


Figure 4.3c : Gestion des PABX.

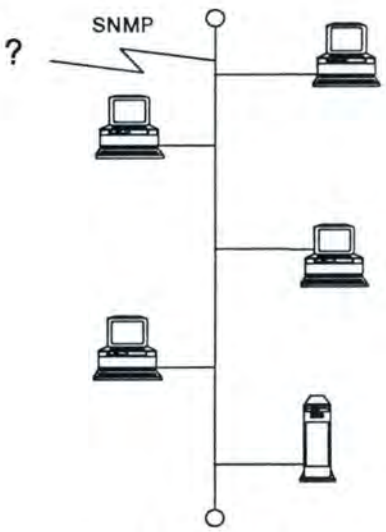


Figure 4.3b :Gestion des LAN.

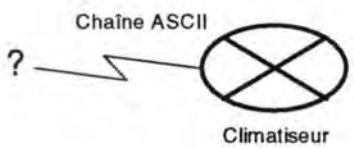


Figure 4.3d : Gestion de la climatisation.

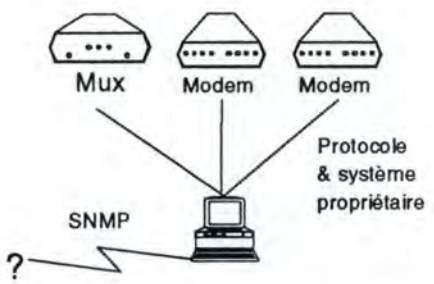


Figure 4.3e : Gestion des modems.

* RPC = Remote Procedure Call : appel de procédure à distance. La procédure n'est pas exécutée localement mais est exécutée d'une manière transparente sur un site client distant.

4.2. Projet d'intégration des gestions.

Nous avons choisi de décrire l'implantation d'un SIGR en suivant le modèle du processus de décision décrit par Mintzberg. C'est donc en reprenant les différentes étapes de ce modèle que nous allons décrire l'intégration des systèmes de gestion de réseau(x) et d'autres systèmes. Avant, nous en rappellerons les étapes et verrons comment elles se traduisent dans le type de projet qui nous intéresse. Ensuite, nous étudierons de plus près chaque étape. Tout au long de cette étude, nous illustrerons nos propos au moyen du cas M.B.B.

4.2.1. Modèle du processus décisionnel et intégration.

- La **première étape** est la prise de conscience de l'existence d'un problème ou d'une opportunité. Cela se traduit habituellement par une *prise de décision de la direction* de l'entreprise. Elle situe le problème et est déjà le fruit d'une prise de conscience préalable ainsi que d'une première analyse complète de la situation dans l'entreprise. La décision de la direction fixe le cadre dans lequel va s'inscrire le projet. Elle peut être formalisée au moyen de règles qui seront utilisées ultérieurement dans l'analyse des solutions.

- La **deuxième étape** consiste en un *diagnostic de la situation* existante. Bien que cela ait déjà été fait en partie pour réaliser la première étape, il s'agit ici d'une étude plus détaillée. On questionnera les opérateurs et les administrateurs des systèmes existants afin de déterminer ce qui est fait ainsi que ce qu'il serait nécessaire d'ajouter pour répondre aux attentes exprimées par la direction.

- La **troisième étape** est la conception d'une *nouvelle solution* ***abstraite,*** c.-à-d. la constitution du cahier des charges reprenant les critères à respecter. Cette étape est l'une des plus importantes. Outre la solution technique, c'est également la solution organisationnelle qui est recherchée.

- La **quatrième étape** est la recherche des *solutions existantes* sur le marché.

- Au cours de la **cinquième étape**, on *analyse* chaque produit proposé par les vendeurs dans le cadre du cahier des charges. Il s'agit en quelque sorte de faire passer le système analysé au travers d'un processus afin d'en obtenir un profil. L'idéal est de procéder par "benchmark[*] ".

- C'est dans la **sixième étape** que nous procédons au *choix* d'un système. Ce choix est alors soumis à la direction pour obtenir l'autorisation de l'implémenter.

- Au niveau de la gestion intégrée, il n'existe pas encore de système prêt à l'emploi (plug and play). Dès lors, si nous sommes autorisés à implémenter un système, il faudra *construire la solution* correspondant à notre société sur base du SIGR choisi: c'est la **septième étape**. Nous recommandons d'installer successivement les fonctionnalités de gestion nécessaires à la gestion des divers éléments à gérés. On travaillera éventuellement suivant un processus d'implantation en "oignon".

- Enfin, il faudra veiller à ce qu'à l'issue de ce processus l'ensemble du personnel (opérateurs, administrateurs et développeurs) soit formé et que l'infrastructure soit en place.

Remarquons qu'outre le modèle du processus de décision, nous allons retrouver le modèle d'apprentissage. En effet, les deuxième et troisième étapes vont constituer l'apprentissage entre utilisateurs et concepteurs de la solution. L'analyse détaillée au moyen de "benchmarks" va introduire l'apprentissage des systèmes informatiques par les concepteurs et les utilisateurs; la septième étape fermera la double boucle.

4.2.2. Première étape : décision de la direction.

Comme nous l'avons sous-entendu ci-dessus, la décision de la direction est elle-même issue d'un processus de décision. En effet, pour l'une ou l'autre raison, la direction a été sensibilisée à la problématique de la gestion et a sans doute dressé un état de la question au sein de la société. Forte de ses observations, elle a édicté des règles de gestion des réseaux qui désormais doivent être prises comme lignes de conduite pour atteindre un objectif décrit en termes de qualité de services. Ces directives seront à la base de la recherche d'une solution par le CTI. Eventuellement, la direction du département des ressources informatiques émettra elle-même des directives concrétisant davantage celles de la direction de la société.

[*] Benchmark = test de matériel (logique ou physique) basé sur une utilisation du matériel dans un environnement proche du monde réel.

Dans le cas de M.B.B., on peut illustrer cela comme suit: au cours d'une réunion avec les directeurs de succursale provinciale, certains problèmes sont apparus, par exemple, à propos des accès aux données du siège central à partir du LAN ou via les modems. De plus, lorsqu'on veut modifier la configuration du LAN, par exemple pour ajouter une application, il faut soit compter sur l'expérience de l'employé ayant en charge la gestion journalière du LAN soit attendre le passage d'un technicien de maintenance attaché au siège central. Ce même problème se pose lorsque le PABX doit faire l'objet d'une modification quelconque. A noter que le problème est d'autant plus important qu'un seul technicien n'est pas capable d'exécuter les deux tâches. Enfin, lors de fortes chaleurs, il arrive que le système de climatisation soit défectueux et qu'il faille envoyer quelqu'un depuis le siège central pour diagnostiquer le problème vu que l'accès à la salle des ordinateurs est interdit au personnel non spécialisé. Le problème de la climatisation n'est détecté que suite à des problèmes avec le WAN et suite aux plaintes du personnel qui utilise le PABX et le LAN.

Conséquemment à ces observations, il est décidé de charger le responsable du CTI d'une recherche d'une solution visant à atteindre le niveau de qualité fixé en améliorant le service offert aux succursales et à leurs courtiers. L'objectif peut se synthétiser comme suit: faire de l'informatique un moyen efficace et disponible pour le traitement des dossiers en réduisant les temps de réaction et d'intervention nécessaires pour régler les problèmes et en optimisant la configuration.

La qualité de service s'exprimera en des termes tels que:

- ☑ Temps de réponse pour consultation d'un dossier du fichier central < = 2 sec.
- ☑ Disponibilité du réseau pendant les heures de bureau > 98 % en moyenne.
- ☑ Disponibilité du réseau les jours ouvrables entre 18.00 heures et 22.00 heures et le samedi entre 09.30 heures et 17.00 heures > 87.5 % en moyenne.
- ☑ Temps d'installation logique d'un nouvel utilisateur (téléphone, connexion LAN et WAN) ≈ 30 minutes.
- ☑ Délai de mise en place d'un poste de travail physique (installation du PC, des logiciels et du téléphone) dans le cours du jour ouvrable suivant.
- ☑ ...

Le cadre dans lequel la solution doit être trouvée est fixé par les règles suivantes:

- ☑ La charge de travail de gestion des succursales ne doit pas être accrue. Seul un membre de chacune de celles-ci peut être affecté à la gestion de ressources informatiques.
- ☑ Les succursales ne peuvent s'occuper que de la gestion administrative de leur parc et de leurs utilisateurs, ainsi que des manipulations de bas niveau ne nécessitant aucune expérience.
- ☑ Les effectifs de maintenance ne peuvent être augmentés.
- ☑ Le CTI est libre de l'organisation de ses moyens, mais reste limité par son budget de fonctionnement.
- ☑ Toutes les succursales doivent disposer d'une qualité de service équivalente.
- ☑ Lorsque la demande lui en est faite, le CTI doit pouvoir faire rapport de ses activités ainsi que du trafic circulant sur les différents réseaux et, ce, dans les deux jours ouvrables qui suivent. Les données minimales à fournir seront déterminées avec le responsable du CTI.
- ☑ ...

A son tour le responsable du CTI va émettre des règles concrétisant certaines attentes au moyen de termes plus proches du monde informatique.

- ☑ Les sauvegardes doivent être effectuées après 22.00 heures.
- ☑ La prise en charge par le site Back-Up doit être faite pour toute indisponibilité de plus de 3 minutes.
- ☑ Les dérangements doivent être minimisés en temps et en quantité. Tout au plus est-il possible d'admettre une augmentation du temps de réponse lorsque des problèmes techniques se posent durant les heures de bureaux.
- ☑ ...

4.2.3. Deuxième étape : diagnostic de la situation.

Cette étape est d'importance majeure et, ce, particulièrement s'il s'agit d'intégrer des systèmes de gestion préexistants. On procèdera au cours de cette phase à une interview des techniciens, des opérateurs et des administrateurs des différents systèmes de gestion existants. Le but est de déterminer les fonctions d'administration et de les structurer en domaines fonctionnels composés de fonctions qui elles-mêmes se subdivisent en activités [BULL94]. Le domaine fonctionnel d'exploitation des serveurs comprend, entre autres, la fonction de gestion des impressions qui à son tour se compose de plusieurs activités dont la manipulation des imprimantes et des consommables. Toutes ces fonctions d'administration seront quantifiées au moyen de leur fréquence (par semaine, par exemple) et on en retiendra également le niveau d'exécution (opérateur du siège central, responsable local, administrateur , ...).

Dans le cas de M.B.B., cette étape pourrait produire une liste dont celle ci-dessous pourrait être extraite; elle concernerait les activités de gestion du LAN.

Fonctions d'administration **Domaines fonctionnels d'administration** Fonctions Activités	Localisation NN = central PP = provincial NP = central assisté du niveau provincial PN = provincial assisté/contrôlé par le niveau central.	Fréquence action(s) / semaine (au niveau de la société)
Assistance Fonctionnelle LAN	PN	50/1
Assistance Technique (interventions)LAN	NP	30/1
Installation matériel et logiciel de base LAN		
Installation matériel de base	NN	90/52
Service de nom	NN	110/52
Applications réseau	NN	20/52
Installations des périphériques - hors imprimantes	NN	55/52
Installations des upgrades logiciels	NP	75/52
Mise en exploitation des applications LAN		
Diffusion d'une version logicielle d'une application	NP	36/52
Autres fonctions	NN	
Gestion des utilisateurs LAN	PN	45/1
Gestion du parc des PC		
Gestion physique	PN	5/1
Gestion technique	NN	5/1
Gestion administrative	PN	9/1
Exploitation des serveurs LAN		
Gestion de l'environnement d'exécution	NP	1/1
Gestion des sauvegardes		
Manipulation des médias	PP	45/1
Autres activités	PP	45/1
Gestion des impressions		
Manipulation imprimante et consommables	PP	50/1
Autres activités	PP	9/1
Automatisation		
Suivi de l'exploitation	NN	45/1
Surveiller l'exploitation des systèmes		
Toutes fonctions	N	90/1
Maintenance corrective Application LAN	NN	1/1
Maintenance corrective Configuration LAN	NN	18/52
Gérer les bases de données locales		
Toutes fonctions	PN	27/1
Surveiller les performances systèmes		
Toutes fonctions	PN	45/1

Optimiser		
Contrôle les configurations et la disponibilité	NN	18/52
Surveillance du système	NN	18/52
Etudes statisitiques	NN	27/52

On procèdera de la sorte pour toutes les fonctions de gestion des différentes ressources de la compétence du CTI (WAN, Hôtes, Modems, multiplexeurs,...).

4.2.4. Troisième étape : conception d'une nouvelle solution abstraite.

Outre l'étude des fonctions déjà assurées, il est nécessaire de recueillir de l'information sur les nouvelles fonctions permettant d'améliorer la qualité du service et/ou d'atteindre les objectifs fixés. Ces informations pourront être récoltées auprès de sociétés qui utilisent un matériel semblable au nôtre et/ou auprès des constructeurs. On tentera de les quantifier suivant la même échelle de fréquence que celle retenue à l'étape précédente.

En ce qui concerne la gestion des PABX dans le cas de M.B.B., on pourrait avoir :

Fonctions d'administration **Domaines fonctionnels d'administration** Fonctions Activités		Fréquence action(s) / semaine (au niveau de la société)
Gestion des utilisateurs		
Définition de nouveaux utilisateurs		2/1
Modification de statut		5/1
Gestion du matériel		
Trouble ticketing		15/1
Gestion du calendrier de maintenance		10/1
Gestion des inventaires		5/1/
Gestion de la configuration		
Test de ligne		100/1
Modification des paramètres		100/1
Visualisation des paramètres		100/1
Surveillance du trafic		250/1
Gestion de la comptabilité		100/1

Notons qu'il n'est pas question ici de parler de la localisation puisqu'à ce stade, cela n'existe pas. On peut envisager de donner une indication du niveau de réalisation préconisé.

L'ensemble des activités et fonctions recensées dans les deuxième et troisième étapes vont nous permettre de dresser un portrait fonctionnel complet du SIGR. Cette liste d'activités va être la base des deux étapes suivantes.

Tout d'abord, il va falloir trouver une solution organisationnelle adéquate face au cadre de travail imposé par la direction. En d'autres termes, il va s'agir de déterminer qui (notion d'organisation) fait quoi (fonction d'administration et objets gérés) et où (notion de localisation, de centralisation) [BULL94]. Ici, on peut envisager la création de nouveaux niveaux de gestion. Plusieurs scénarios vont probablement se dessiner; c'est sur base du cadre d'exécution fixé par la direction qu'une première sélection va s'opérer.

Chez M.B.B., on proposera par exemple les scénarios suivants* en se basant sur les diverses fonctions d'administration et en les répartissant.

1. Attribuer la gestion opérationnelle des LAN, des PABX , de la climatisation et des modems aux succursales.
   Laisser la gestion du WAN au CTI.

* Pour plus de lisibilité et de concision, nous ne les exprimerons que sous une forme synthétique. En principe, cela s'exprimerait également au moyen de la liste des activités de gestion (anciennes et nouvelles) portant mention des niveaux organisationnels d'attribution de la fonction.

Attribuer le développement d'applications et la conception au CTI.
Constituer un pool de réparateurs pour l'ensemble du matériel et le répartir dans les succursales.

2. Décharger au maximum le niveau provincial, en ne lui laissant plus que la charge des manipulations quotidiennes obligatoires (Back-up, consommables, ...).
Créer un réseau de gestion des télécommunications entre les succursales et le siège central.
Répartir les techniciens sur le territoire et accroître leur polyvalence.

La seconde partie consistera à se donner les moyens correspondant aux scénarios retenus. Pour ce faire, dans chaque fonction d'administration, nous allons rechercher comment implémenter les directives. Cela se concrétisera par des critères auxquels devra répondre le SIGR. On se basera sur la littérature et sur toute autre source d'informations (par exemple : l'expérience d'autres sociétés diffusée sur Internet). On terminera cette étape par une spécification de ces critères en des termes techniques qui prendront en considération les attentes que l'entreprise a du système. Nous aurons ainsi constitué le cahier des charges technique. On obtiendra le cahier des charges complet en y adjoignant les critères non techniques tels que la possibilité de disposer d'une aide 24 heures sur 24, le nombre de systèmes vendus (réputation), la maturité du système ,...

A ce niveau, les critères extraits de la liste des fonctions d'administration chez M.B.B. pourrait ressembler à celle-ci [BULL94]:

Tableau des critères relatifs à l'exécution des grandes fonctions de gestion.		
Grandes Fonctions	**Critères**	**Abréviation**
Assistance	temps de réponse (entre signalement et résolution)	temps de réponse
	couverture de l'assistance (fonctionnel et technique) et souplesse de l'assistance	couverture
	capacité d'automatisation des actions d'assistance	automatisation
	retour d'expérience (capacité à utiliser, engranger au mieux le fruit des actions d'assistance passées)	capitalisation
Installation	réplication d'une action sur plusieurs cibles, automatisation des actions d'installation et couverture des cibles	réplication
	rapidité et capacité d'installation en dehors des heures de travail	rapidité
	minimisation des coûts d'installation	coût
	capitalisation de la compétence d'installation et retour d'expérience (capacité à utiliser, engranger au mieux le fruit des actions d'assistance passées)	capitalisation
Exploiter les serveurs	répétition d'une action sur plusieurs cibles de nature différente	répétition
	garantie dedisponibilité des ressources, et capacité d'optimisation de la maintenance corrective et préventive	disponibilité
	capacité d'exploitation de plusieurs serveurs simultanément	exploitation
	retour d'expérience (capacité à utiliser, engranger au mieux le fruit des actions d'assistance passées)	capitalisation
Surveillance	disponibilité garantie des informations de management issues de tout et des parties des systèmes, pouvoir corréler des informations issues de plusieurs sites	supervision
	possibilité d'actions sur des cibles de un ou plusieurs sites	droits d'action
	retour d'expérience (capacité à utiliser, engranger au mieux le fruit des actions d'assistance passées)	capitalisation
Intégration	capacité d'accès à la connaissance des systèmes et à la vision générale, capacité de réalisation de l'intégration des systèmes, et capacité de réunion de l'expertise des différents domaines concernés par l'intégration (en particulier des applications)	compétence
	capacité de mise à disposition de plates formes d'intégration	moyens
	capacité d'appréhension des différences locales	vision
	retour d'expérience (capacité à utiliser, engranger au mieux le fruit des actions d'assistance passées)	capitalisation
Diffusion	capacité de diffusion de tout objet sur toute cible	droits sur objets
	capacité de diffusion quelque soit l'état spécifique de la cible (place disque, processus actif...), capacité à savoir si le système récepteur possède une configuration favorable (version précédente, cohabitation de version) favorable	vision cible
	capacité de prise de connaissance de l'état de la diffusion, capacité de contrôle de la cohérence de la diffusion	cohérence
	retour d'expérience (capacité à utiliser, engranger au mieux le fruit des actions d'assistance passées)	capitalisation
Gestion utilisateurs	capacité de diffusion des droits inter-entités administratives	droits sur objets
	rapidité de délivrance des droits à un utilisateur	rapidité
	sécurité du système de gestion des utilisateurs	sécurité
	retour d'expérience (capacité à utiliser, engranger au mieux le fruit des actions d'assistance passées)	capitalisation

Au point 4.3, nous verrons plus en détail les critères recommandés dans la littératture ainsi que ceux que nous préconisons.

Pour être complet, il faut ajouter qu'à ce stade, il peut apparaître que la solution optimale soit l'externalisation de la gestion. Dans ce cas, il faut veiller à garder la mainmise sur certaines choses considérées comme stratégiquement importantes. A titre d'exemple, citons le "help desk" qui doit rester au sein de l'entreprise si elle veut connaître les problèmes auxquels elle est confrontée. Bien qu'aujourd'hui les entreprises de service de gestion soient rares en Belgique, on peut penser que suite à la libéralisation des télécommunications, notre pays verra naître de telles sociétés.

4.2.5. Quatrième étape : recherche des solutions existantes.

Au cours de cette étape, il s'agira d'établir la liste des divers produits du marché qui sont susceptibles de répondre à notre attente. Le choix de ceux-ci pourra avoir été restreint par certains critères non techniques repris dans le cahier des charges. On se constituera cette liste au moyen de la littérature et de l'information obtenue à l'extérieur (consultance, autres sociétés, ...).

Dans le cas de M.B.B., on peut supposer que ce sont les systèmes étudiés au chapitre 3 qui sont retenus.

4.2.6. Cinquième étape : analyse des solutions.

Fort des informations obtenues à l'étape précedente, nous nous faisons présenter les produits des divers constructeurs et vendeurs. Au préalable, nous leur avions remis notre cahier des charges afin qu'ils nous présentent les produits les plus à même de répondre à nos besoins. Nous procéderons alors à une analyse en deux phases séparées par un premier choix.

Tout d'abord, chacune des réponses fera l'objet d'une analyse détaillée. Pour réaliser celle-ci, nous procéderons préalablement à une pondération des divers critères afin d'accentuer l'adéquation avec l'objectif fixé. Les poids attribués aux différents critères s'appuieront, par exemple, sur la fréquence des activités à l'origine de ces critères ou encore sur la structure de l'organisation de gestion qui pourra exiger ou non un SIGR distribué ou orienté client/serveur [WIER93].

Ensuite, nous retiendrons deux ou trois systèmes que nous implémenterons sous forme de prototypes dans un environnement quelconque mais uniforme; idéalement, il constituera un sous-ensemble de nos réseaux. Cette façon de faire doit nous permettre de vérifier les informations fournies dans l'offre du constructeur et d'acquérir une meilleure connaissance des possibilités des systèmes ainsi testés.

S'il apparaît qu'aucun des systèmes retenus n'offre des perspectives de réponse satisfaisante à nos attentes, nous pouvons alors procéder à de nouveaux "benchmarks" sur d'autres SIGR préalablement retenus ou alors assouplir certains critères non techniques afin d'élargir le champ de sélection. Par exemple, on pourrait laisser tomber le critère de représentation nationale d'un constructeur. Si d'autres systèmes sont proposés, on recommencera le processus d'analyse.

Chez M.B.B. nous pouvons supposer que deux produits ont été retenus. Ce sont YYY et XXX.

4.2.7. Sixième étape : choix de la solution.

Sur base des résultats obtenus lors de l'analyse, on choisira un système ou on décidera de remettre à plus tard l'acquisition d'un tel produit. Le produit choisi respectera nos critères de sélection pondérés. Il faudra veiller à son ouverture et à ses perspectives d'avenir. En effet, le monde des télécommunications est en perpétuelle évolution et la gestion des réseaux est en passe de s'étendre à la gestion des systèmes (ensemble d'applications collaborant à l'aide du/des réseau(x)) ainsi qu'à la gestion des bases de données. On en arrive à des systèmes de gestion de l'entreprise qui nécessitent une ouverture maximale vers l'extérieur tant la multitude et la diversité des objets gérés sont grandes.

M.B.B. aura choisi **XXX** pour son ouverture et sa souplesse; en plus, ce dernier dispose en standard des modules de communication correspondant à ses besoins. De faibles efforts de développement semblent nécessaires. Enfin, l'expérience d'installation existe en Belgique puisque ce produit a déjà fait l'objet d'une installation auprès d'une grande banque.

4.2.8. Septième étape : adaptation de la solution.

il n'existe actuellement pas de produit prêt à l'emploi au niveau de la gestion de réseau(x) dans la hiérarchie TMN. Lors de l'étape précédente, nous avons choisi une plate-forme que nous allons adapter à nos besoins et à notre configuration. Notons que le terme plate-forme est ambigu puisqu'il désigne à la fois la base pour la construction d'un SIGR et une structure particulière de SIGR. Pour plus de clarté, nous emploierons le terme *base de travail.*

La *base de travail* va faire l'objet d'une adaptation plus ou moins importante suivant la complexité des environnements dont on souhaite intégrer la gestion et suivant les fonctionnalités offertes par le système en standard. Nous recommandons d'avoir un attribut de sélection à forte pondération pour la facilité de personnalisation du SIGR. C'est non seulement au cours de son installation, mais aussi tout au long de sa vie qu'il faudra faire évoluer la base de travail avec les systèmes de l'entreprise.

L'implantation chez M.B.B. est faite par du personnel de chez XXX qui collabore avec certains opérateurs (ceux qui ont procédé aux benchmarks et qui seront désignés comme développeurs). Pendant ce temps, on procédera à la formation des autres opérateurs et des administrateurs.

A l'issue de cette étape, le système doit être opérationnel. Cela signifie que, outre l'adaptation technique, le personnel de gestion doit avoir été formé, l'infrastructure de travail doit avoir été adaptée aux nouvelles tâches et à la nouvelle organisation et enfin l'agenda des activités planifiées (scheduler) doit avoir été établi. Pour que

cela soit possible, il faut avoir établi un plan de formation, d'installation et d'opération au cours des phases précédentes.

4.3. Critères d'analyse .

Comme nous l'avons déjà dit lors de l'établissement du cahier des charges, outre les critères déterminés sur base de la liste des fonctions d'administration et des directives de la direction (au point de vue de la qualité de service et du cadre de référence), nous retiendrons aussi les critères cités par la littérature et d'autres organisations. Nous commencerons ce point par une énumération des divers critères que nous avons trouvés au cours de nos recherches; après cela, nous tenterons de décrire les critères retenus dans le cas de M.B.B. aussi complètement que possible.

4.3.1. Liste des critères.

Les critères que nous présentons ici sont des critères génériques d'appréciation et de comparaison. Lorsqu'il s'agit d'évaluer un système, on aura au préalable spécifié les exigences des systèmes existants (par exemple : les protocoles de communications et de gestion utilisés ou utilisables pour le matériel existant). Nous recommandons la prise en compte des différents points repris ci-dessous lors de la rédaction d'un cahier des charges techniques.

a. **Architecture** : à ce niveau, on veillera particulièrement à la concordance avec la structure de référence proposée par la recommandation M.3010 de UIT-T.

- Modularité [WIER93] : cet important critère est synonyme de facilité de remplacement des composants logiciels dépassés et/ou d'ajout de nouveaux. Le développement au moyen d'une technique orientée objet permet d'offrir une bonne modularité.
- Distribution [WIER93] : cela permet une décentralisation des fonctionnalités tout en présentant un aspect intégré.
- Client/serveur [FCEM94] : cette technique permet, par exemple, d'ajouter aisément une nouvelle station de gestion. De plus, si l'aspect d'interfaçage est implémenté sur chaque station, cela permet de limiter le trafic sur le réseau puisque seules les informations seront envoyées. Si une telle architecture est proposée, il faudra tenir compte d'autres critères en résultant:
  - Nombre maximum de clients.
  - Cause de la limitation du nombre de clients (serveur, matériel, ...).
  - Règles de concurrence entre clients.
  - Possibilité de multi-traitements.
  - .... .
- Opérateurs multiples : l'organisation et l'architecture client/serveur permet en principe d'avoir des opérateurs multiples, mais comment sont-ils gérés?
- Organisation [BBL94] : étant donné une structure très complexe, le SIGR peut n'intégrer qu'une partie des systèmes et un SIGR "maître" (≈ MoM) supervise leurs activités. Ou alors, plusieurs SIGR peuvent coopérer (≈"Peer-to-Peer").
- Tolérance aux fautes [FCEM94] : nous l'avons dit et répété, les télécommunications et l'informatique en général sont des ressources critiques pour les entreprises d'aujourd'hui; a fortiori, il en sera de même des SIGR qui en assurent le bon fonctionnement.

b. **Offre d'API (Application Programming Interface)** [FCEM94] : les API sont un moyen pour assurer la flexibilité et l'extensibilité du SIGR. Grâce à elles, l'accès aux informations du SIGR est possible pour des applications étrangères au système. Ces mêmes applications peuvent également par ce moyen fournir au SIGR des informations adaptées.

c. **Système expert**[WIER93] : étant donné la complexité des environnements, il est pratiquement impossible pour un humain d'avoir une bonne connaissance des divers systèmes gérés. Un système expert semble être un composant important qui permet la corrélation des évènements perçus dans les divers réseaux et systèmes.

- Support pour trouble tickets (rapports de problème) [FCEM94] : il s'agit d'engranger par ce moyen plus de connaissances pour venir en aide aux opérateurs confrontés à un problème et/ou alimenter la base de connaissances du système expert.
- Types d'actions [BBL94]: le système expert peut produire différents effets. Il pourra déclencher l'exécution d'une fonction, simplement donner des directives à l'opérateur en allant rechercher le(s) trouble-ticket(s) ad-hoc, proposer une action ou séquence d'actions qu'il n'exécutera qu'avec l'aval de l'opérateur, proposer des simulations, détecter proactivement des erreurs, ...
- Auto-apprentissage : le SIGR peut alimenter sa base de connaissances avec les trouble tickets, avec l'ensemble des actions des opérateurs et les réactions induites de l'environnement intégré, ...
- Langage utilisé : certains langages d'écriture des règles d'inférence sont relativement simples et répandus (PROLOG); d'autres le sont moins mais sont censés être plus puissants. Il s'agit de peser le pour et le contre en fonction des connaissances du personnel du CTI.
- Accessibilité : le système peut être "on-line", c.-à-d. inclus dans le SIGR, ou bien "off-line", c.-à-d. exécuté en dehors du SIGR au sens strict tout en ayant des moyens d'interface avec lui.
- Surveillance intelligente [FCEM94] : ce devrait être un des premiers buts du système expert. Il s'agit de ne pas signaler des problèmes résultant logiquement d'un autre situé en amont. Ainsi si un concentrateur A est défectueux, il ne doit pas signaler que le concentrateur B, situé au delà de A, est défectueux alors qu'en fait son état lui est inconnu.

d. **Implémentation de standards** (par exemple : OSF/DME) : nous avons vu que mis à part les organismes de standardisation, il existe d'autres groupes de réflexion comme OMG. Dès lors, si l'on désire que le SIGR respecte leurs recommandations, on le précisera parmi les attributs de sélection.

e. **Installation** : il s'agit bien entendu de l'installation du SIGR , des applications, mais aussi de tout nouveau composant dans un des réseaux gérés.

- Auto-découverte [FCEM94] : le SIGR doit être à même de découvrir d'initiative ou à la demande les composants. Idéalement, il doit pouvoir découvrir les différents protocoles de niveau 3 (IPX, AppleTalk,...).
- Auto-configuration [FCEM94] : la configuration des environnements gérés doit pouvoir être découverte et/ou chargée à partir d'un fichier de configuration.

f. **Domaine d'application privilégié** [WIER93] : souvent, au sein d'une entreprise, il y a un type de réseau qui est plus important ou qu'on désire privilégier. De même, parmi les SIGR existants, il n'est pas rare que le système ait été conçu à l'origine pour la gestion d'un domaine et qu'on l'ait étendu à d'autres. Si tel est le cas, il vaut mieux choisir un système qui a été développé à l'origine pour le type de réseau privilégié.

g. **Intégration multi-vendeurs** : c'est bien entendu le critère majeur de cette liste puisqu'un SIGR doit offrir par essence une telle intégration. Cette dernière sera plus ou moins complète. Ci-dessous, nous reprenons les trois grandes sphères d'intégration.

- Interfaces avec les protocoles de communication: on définit à ce niveau les possibilités de connexion avec les composants contrôlés. A titre d'exemple nous pouvons citer TCP/IP et X.25.
- Interfaces avec les protocoles de gestion [WIER93]: nous avons vu qu'il existait deux grandes familles de protocoles de gestion. Ici, on précise celui ou ceux dont l'utilisation est requise. Notons qu'on désirera souvent que le SIGR puisse recevoir des messages ASCII (formatés ou non).
- Emulation de terminal offerte[BBL94]: de manière à pouvoir lancer des sessions sur les systèmes existants, il est utile de préciser les émulations de terminal désirées.

h. **Gestion de données** [WIER93]: de nombreux fichiers font souvent partie du SIGR (base de données de gestion, fichiers de trouble tickets, base de connaissances,...). Les données de gestion seront souvent organisées dans une MIB (Management Information Base) OSI ou TCP/IP; il est surtout intéressant de savoir si le système de gestion sous-jacent est compatible avec celui utilisé

dans l'entreprise. La société dispose alors d'une bonne connaissance du SGBD, ce qui en facilite la manipulation.

i. **Performances du système** [WIER93] [BBL94]: s'il est intéressant d'être compatible avec de nombreux systèmes, il faut avant tout que les performances du SIGR soient suffisamment bonnes pour inciter les opérateurs à l'utiliser. Celles-ci sont souvent exprimées au moyen de :

   - nombre maximum d'évènements pris en charge par seconde (en moyenne et en pointe),
   - traitement des excédents.

j. **Facilité d'emploi**[WIER93]:

   - Standard d'interfaçage: on précise ici le(s) standard(s) d'interface accepté(s) (OSF/Motif, Windows, X Window, ...
   - Interface Homme Machine: la présentation des informations et du système intégré doit permettre une navigation aisée parmi les composants. On peut penser présenter les informations sous une forme hiérarchique avec des possibilités de zoom, de représentation des sites sur cartes, ...
   - Personnalisation: ce que chaque utilisateur peut adapter pour sa facilité de travail sera ici pris en considération; on définira aussi ce qui est personnalisable à l'installation et/ou en cours d'utilisation par l'administrateur (par exemple : les alarmes et leur notification). Remarquons que ce critère de personnalisation peut être pris en compte à d'autres niveaux que l'interfaçage; ainsi, un utilisateur doit pouvoir par exemple modifier la fréquence d'interrogation (polling) pendant sa session; la fréquence définie par l'administrateur reste valable pour d'autres sessions.
   - Opérateurs multiples [FCEM94]: lorsqu'un ensemble de systèmes complexes est géré, on répartira éventuellement la charge de travail parmi plusieurs opérateurs. Tous n'auront pas les mêmes possibilités de regard sur l'ensemble des systèmes. Il n'est par exemple pas utile pour un opérateur chargé de la gestion d'un WAN sur lequel sont connectés des LAN, d'avoir une vue des composants du LAN. Dans ce cas, il s'agit de pouvoir distinguer différents types d'opérateurs.
   - Formation du personnel: nous avons déjà cité le problème dans le chapitre 3. Rappelons que celle-ci peut nous donner une idée de la complexité du système mais qu'elle reste de toute façon un facteur psychologique important pour le personnel devant être formé et recyclé.
   - Messagerie et mémo: il est utile de pouvoir disposer d'un bloc-notes pour chaque site et/ou pour chaque élément géré; de cette façon, chaque opérateur accédant à un site peut prendre connaissance de notes et commentaires laissés par un autre ou par lui-même. Une durée de vie peut être associée à chacune de ces informations.
   - Documentation: la qualité de la documentation sur papier et "on-line" mise à disposition par le constructeur ainsi que les informations circulant au sujet du SIGR seront à apprécier.

k. **Outils de logging** (enregistrement) [BBL94] : il est souhaitable de garder trace des évènements, messages et actions. Outre la constitution d'une base de connaissances, nous pouvons citer la nécessité de faire des rapports (reporting) et de les personnaliser [FCEM94]. Le but est de pouvoir étudier le comportement des systèmes gérés d'une manière stratégique et tactique afin d'en tirer des conclusions quant à la configuration, ...

l. **Analyse des performances** [AETC93] [BBL94] : hormis l'analyse a posteriori, il peut être utile de pouvoir observer le comportement des systèmes en temps réel. La manière la plus appropriée sera certainement la présentation sous forme de jauges et de compteurs.

m. **Gestion du SIGR** [WIER93] : mise à part la gestion des systèmes intégrés, nous devons considérer les moyens donnés à l'administrateur du SIGR pour en assurer la gestion.

   - contrôle d'accès : quels sont les moyens disponibles au niveau de la sécurité pour contrôler les accès au SIGR et à ses fonctions?

- degré de contrôle [FCEM94] : le SIGR doit permettre à tout utilisateur en possédant les droits d'accéder à toutes les informations et fonctions utiles pour le contrôle des ressources gérées. Cela sous-entend la possibilité de modifier la configuration des objets contrôlés.
- contrôle de capacité : quelles sont les possibilités de définition de niveau d'accès aux fonctions et informations du SIGR?
- back-up : en cas de problème, un système de back-up doit permettre la reconfiguration complète du SIGR et des systèmes gérés. La présence d'une station de gestion (ou serveur) de secours est recommandée.

n. **Possibilités d'extension** : on sera attentif au fait que le SIGR peut servir pour la gestion de systèmes et BD en y intégrant des modules logiciels disponibles.

o. **Infrastructure matérielle (hardware)** [WIER93] : on observera ici la configuration matérielle recommandée pour faire tourner le SIGR et/ou sa compatibilité avec les moyens disponibles.

p. **Coûts**[WIER93] : nous attirons l'attention sur le fait qu' une solution peut au premier abord paraître financièrement intéressante mais qu'il faut également tenir compte du coût des mises à jour et des personnalisations.

q. **Fournisseur**[WIER93] et **service à la clientèle** [FCEM94] [BBL94] : l'implantation sera plus que probablement faite par le fournisseur et c'est lui qui devra en assurer la maintenance et le suivi. Dès lors, le sérieux, l'expérience et le savoir faire d'un fournisseur sont autant de gages de réussite.

r. **Statut du SIGR** [WIER93] : il faut tenir compte ici de la maturité du produit proposé.

4.3.2. Recommandations techniques pour M.B.B. .

Ce point donne un exemple de recommandations techniques réalisées sur base des critères relevés au point précédent. Nous spécifierons éventuellement les raisons des choix faits au moyen de caractères italiques.

a. **Architecture** :

- La préférence sera donnée à un système client/serveur permettant la connexion simultanée de plusieurs opérateurs. Idéalement le SIGR permettra le traitement parallèle de requêtes ne portant pas sur le même domaine. Les règles de gestion de la concurrence seront précisées par le fournisseur. Elles pourront éventuellement faire l'objet d'une redéfinition par le CTI de M.B.B..
- Pour l'adjonction et la modification des modules logiciels, le système supportera la programmation en C++ ou éventuellement en C. *Il s'agit là d'un standard de la programmation orientée objet, gage de modularité.*
- Le SIGR permettra l'installation ultérieure d'une station de gestion maître devant permettre la supervision du travail des opérateurs. Une connexion vers un autre serveur de sauvegarde doit être possible. Cette connexion ne peut être permanente et ne doit être ouverte qu'à la demande d'un administrateur ou automatiquement lors de procédures de sauvegarde. Le système offrira une bonne tolérance aux fautes.

b. **API** :

- Le SIGR devra offrir une compatibilité avec les API classiques tels que XMP. *Ceci permet de réduire le travail des développeurs puisque des applications utilisant ces API existent.*
- Le SIGR fournira les API nécessaires à la personnalisation de la base de travail.
- Les API seront de préférence écrit en C ou C++.

c. **Système expert** :

- Un système de trouble ticketing sera fourni avec le SIGR. Ce système pourra être alimenté manuellement par les opérateurs ou automatiquement lors de la détection d'un problème.
- Le SIGR sera doté d'une capacité d'auto-apprentissage basé sur ses actions ainsi que sur celles des opérateurs. Cette base de connaissances doit permettre une surveillance intelligente des réseaux.

- Les règles de la base de connaissances seront écrites en PROLOG ou tout langage de haut niveau. L'existence d'une interface permettant l'écriture de règles en langage naturel est considérée comme un atout. *Le but est de permettre un encodage des connaissances de tous les opérateurs et des experts réseaux le plus aisément possible.*
- Le système expert fera automatiquement une corrélation entre les nouveaux évènements et ceux stockés en mémoire. Il fera des mises à jour automatiques des connaissances déduites de son raisonnement. Il déclenchera des actions en temps opportun, c.-à-d. en fonction de ses connaissances et des règles qu'il possède.
- Le système expert permettra quatre types d'actions : réactions automatiques basées sur des scénarios, propositions de réactions que les opérateurs doivent avaliser, propositions de réactions à exécuter manuellement et simulations d'exécution de tout scénario à la demande.
- Le système expert sera "on-line". *Etant donné la diversité des réseaux à gérer et la faible polyvalence des opérateurs, nous préconisons un système on-line surveillant les réseaux en permanence.*

d. **Installation** :

- le SIGR devra pouvoir découvrir tous les équipements gérés quels qu'en soient le constructeur et le protocole utilisé. De même, la configuration des systèmes gérés sera chargeable à partir des fichiers présents dans les systèmes de gestion existants et à partir de disquettes DOS ou de bandes de type "tape streamer".

e. **Domaines de gestion** :

- le SIGR devra être capable de gérer simultanément:
  - des LAN Novell NETWARE,
  - des PABX compatibles avec SNMP ayant leur propre MIB,
  - un WAN utilisant un système de gestion propriétaire basé sur des RPC,
  - des modems et multiplexeurs compatibles SNMP,
  - des appareils de climatisation pouvant émettre et recevoir des messages ASCII formatés sur une porte série asynchrone.

Ces précisions se justifient au vu du point 4.1.3.(infrastructure de gestion existante).

f. **Intégration** :

- Interfaces de communication et suites de protocoles supportées:
  - ⇒ LAN = TCP/IP grâce au NLM de la version 4.01 de Novell NETWARE et Novell NETWARE.
  - ⇒ WAN = TCP/IP.
  - ⇒ Modems = TCP/IP.
  - ⇒ PABX = X.25.
  - ⇒ Climatisation = RS232.
  - ⇒ La possibilité de connexion au moyen des protocoles OSI est considérée comme un atout.
- Interfaces de gestion :
  - ⇒ LAN = SNMP grâce au NLM de la version 4.01 de Novell NETWARE.
  - ⇒ WAN = propriétaire.
  - ⇒ Modems = SNMP.
  - ⇒ PABX = SNMP.
  - ⇒ Climatisation = propriétaire.

⇒ La possibilité d'avoir une interface avec CMIP et/ou SNMPv2 est considérée comme un atout.

- Emulation terminal :

  ⇒ Emulation d'affichage PC - pour l'accès direct aux serveurs des LAN.

  ⇒ Emulation de terminal vt100 pour le lancement de session directement au niveau du mainframe.

g. **Gestion de données** :

- L'ensemble des bases de données seront exploitables à partir d'un SGBD relationnel de type SQL compatible avec EGG. *Nous avons supposé que M.B.B. utilise le SGBDR EGG pour la gestion de ses bases de données.*

h. **Performances du système** :

- Le SIGR permettra au moins le traitement de 10 évènements par seconde en moyenne et 100 pendant de courtes pointes n'excédant pas 5 secondes. Au-delà de ce nombre, seuls l'adresse et l'identifiant de l'information seront conservés pour un "polling" ultérieur.

i. **Facilité d'emploi** :

- L'interface utilisateur sera de type graphique.
- L'administrateur définira les filtres pour interpréter les divers évènements et spécifiera les alarmes correspondantes.
- L'administrateur pourra associer des scénarios aux diverses alarmes. Ces scénarios comprendront entre autres le routage qui doit leur être appliqué (écran, téléphone vers un sémaphone, imprimante, ...).
- L'administrateur aura la possibilité de rendre les alarmes compréhensibles, d'en définir la manifestation (couleur, son, ...), le niveau et la priorité. Pour les rendre compréhensibles, il pourra y ajouter des commentaires en français (néerlandais).
- Toutes les définitions d'alarmes, de scénarios, ... se feront au moyen d'un langage déclaratif ou de la technique "pointer cliquer".
- L'administrateur définira les vues possibles. Il pourra le faire en incluant des cartes, des photos au format GIF et des icônes prédéfinies ou créées à l'aide d'un outil graphique fourni.
- Les vues présentées par le SIGR doivent présenter l'état de l'environnement intégré en temps réel.
- Les alarmes et informations pourront être représentées graphiquement ou de manière textuelle.
- Il devra être possible d'obtenir par un double click toutes les alarmes actives pour un site et un composant. Des explications et une liste des trouble tickets assimilés devront pouvoir être aisément obtenues pour celles-ci.
- Les informations des sondes seront visualisées sous forme de graphique(s), compteur(s) et jauge(s).
- Chaque utilisateur pourra ajouter des macros, menus et tout autre élément d'interface afin de se créer un environnement de travail propre. Les informations qu'il organisera dans ses écrans devront toujours rester cohérentes avec celles décrites par l'administrateur.
- Un mémo attaché à chaque site et consultable en permanence par tous les opérateurs sera un atout.
- Un manuel d'aide en hypertexte ou en hypermédia sera disponible "on-line"

j. **Outils de logging** :

- Toutes les informations circulant sur le réseau de gestion des télécommunications seront stockées. Un outil d'agrégation des informations sera fourni afin de permettre le compactage des données. Le droit d'utilisation de cet outil devra faire l'objet d'une attribution particulière.

- Un outil sera proposé pour l'analyse des données stockées. Cet outil devra permettre la mise en forme de rapports pouvant contenir des illustrations graphiques. La rédaction de ces rapports pourra se faire automatiquement à l'issue d'une période fixée ou à la demande.

  Le CTI doit pouvoir rapidement réaliser des rapports.

- Les données stockées pourront être exploitées à partir de tout SGBDR dont EGG et devront également pouvoir être exportées vers un format DBF sur disquettes DOS.

k. **Analyse des performances** : outre l'analyse différée des évènements, les opérateurs devront disposer d'une "sonde" leur permettant d'analyser les performances de tout composant de l'environnement intégré.

l. **Gestion du SIGR** :

- L'administrateur pourra définir les différentes catégories d'utilisateurs. Pour chacune de celles-ci, il pourra déterminer une plage horaire d'utilisation, les objets ou catégories d'objets accessibles et les manipulations autorisées. Il pourra également permettre certaines modifications de l'environnement de travail propre à chaque utilisateur (non seulement au point de vue de l'interface homme-machine, mais aussi au point de vue de l'intervalle de polling ou de l'ajout d'informations dans la sonde).
- Un système d'aide à l'administration du SIGR sera considéré comme un atout. Ce système conseille l'administrateur pour la définition des droits sur base de paramètres connus (configuration des systèmes) et communiqués (nombre d'opérateurs),.
- Un back-up automatique et paramétrable par l'administrateur (fréquences et données reprises) sera prévu. Plusieurs types de back-up seront possibles (quotidien, hebdomadaire, ...). Ces back-up se feront sur bandes de type "tape streamer" standard.

m. **Compatibilité avec des standards** :

- A brève échéance, le système doit pouvoir être rendu compatible avec OSF/DME.

Conclusion.

Nous pouvons tirer une conclusion de chacune des deux parties de ce travail.

Tout d'abord, il faut remarquer que les SIGR sont à l'intersection du monde des télécommunications et de nombreuses autres branches de l'informatique. Nous avons pu constater que ces systèmes pouvaient également faire l'objet d'une étude aux points de vue de la gestion de projet et de l'organisation de l'entreprise, de l'intelligence artificielle et des systèmes experts, des bases de données et de l'interface homme machine. Pour notre part, nous n'avons qu'ouvert des portes sur ces domaines de recherches, laissant à d'autres le soin de les refermer. Cela nous a permis de constater que les SIGR sont du domaine des télécommunications, mais n'en sont pas de son ressort exclusif. Nous en voulons pour preuve le petit nombre de standards et de protocoles mentionnés.

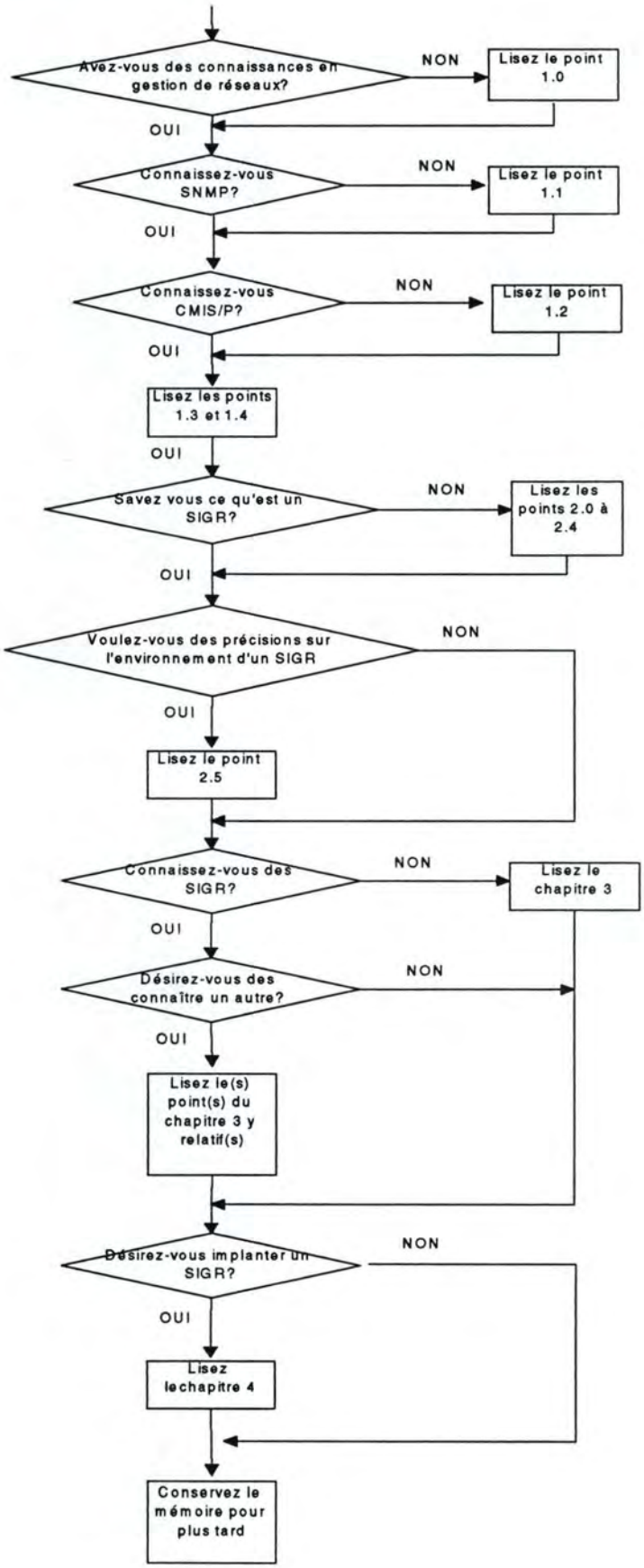
La seconde partie nous amène à constater qu'il y a de très nombreux produits à l'heure actuelle, mais qu'aucun ne semble s'imposer. Certes, HP OpenView semble très répandu, mais il est encore incomplet et des approches telles que celles de Network Managers et Alcatel doivent être prises en considération au moment où l'on dresse la liste des systèmes existants. Actuellement, il n'existe aucune définition officielle des caractéristiques et fonctionnalités de ce type de produits. Cela a pour conséquence qu'un grand nombre de constructeurs prétendent offrir un tel système. La liste de critères proposée au chapitre 4 et la méthode d'analyse en dix points que nous avons faite au chapitre 3 nous semblent être deux moyens utiles pour mettre en évidence les caractéristiques des produits proposés pour faire de la gestion intégrée. Nous insistons sur le fait que cette liste de critères n'est pas exhaustive car chaque implantation doit être précédée d'une analyse détaillée des besoins et attentes de l'entreprise; elle devra permettre de pondérer les critères, d'en rejeter certains et d'en ajouter d'autres. Enfin, il nous est également apparu que l'implantation d'un SIGR est synonyme d'adaptation du produit retenu à l'entreprise (dans la plupart des cas, par un développement de modules logiciels). Il est clair qu'il n'existe aucun SIGR qui puisse être considéré comme prêt à l'emploi (Plug and play); c'est une conséquence de l'absence de standard. L'évolution permanente des télécommunications et cette absence de standard amènent l'acheteur à être attentif aux possibilités d'évolution du produit car c'est l'unique moyen, à l'heure actuelle, de réduire les risques liés à la dépréciation du SIGR.

Que nous réserve l'avenir? Le mouvement d'intégration est lancé. La gestion intégrée de réseaux a créé un courant qui est en passe d'entraîner avec lui la gestion de systèmes et de tout ce qui repose sur un processeur et pouvant être géré à distance. Certains (BULL, HP) ont ainsi déjà lancé l'idée de *systèmes de gestion d'entreprise*. Cela doit permettre la centralisation de la gestion de l'ensemble des moyens informatiques et de télécommunications d'une entreprise. Il faut toutefois veiller à garder une certaine décentralisation de la gestion en répartissant, par exemple, le système intégré de gestion au sein des opérateurs et/ou des sites. Dans ce cas, l'intégration doit permettre une corrélation des évènements et une vue globale des ressources améliorant ainsi les traitements d'erreurs; la distribution permettra le partage des tâches entre les opérateurs. Il faut en effet se rappeler que souvent, l'homme restera l'élément clé. Il semble peu réaliste de penser qu'un même opérateur soit un jour capable de prendre toutes les décisions relatives à la configuration de LAN, de PABX et d'un système d'exploitation.

Il ne suffit pas d'intégrer pour bien gérer.

ANNEXES.

Annexe A : Plan de lecture.



Annexe B : Recommandation M.3010.

PRINCIPES POUR UN RESEAU DE GESTION DES TELECOMMUNICATIONS (RGT).

1. Définition d'un RGT.

Le RGT prend en charge les besoins des Administrations en matière de gestion pour la planification, la mise en oeuvre, l'installation, la maintenance, l'exploitation et l'administration des réseaux et des services de télécommunications. La gestion consiste en un ensemble de possibilités d'échange et de traitement d'informations de gestion, pour aider les administrations à conduire efficacement leurs affaires. Les services et protocoles présentés dans la recommandation X.700 constituent un sous-ensemble des possibilités de gestion pouvant être fournies par le RGT et requises par une Administration.

2. Relation entre RGT et réseau de télécommunications.

Un RGT est un réseau distinct du réseau géré qui assure l'interface d'un réseau de télécommunications en plusieurs point pour envoyer/recevoir des informations à/de ce réseau et contrôler son exploitation. Il se peut que le RGT utilise des parties du réseau géré pour assurer ses propres communications.

3. Exemples de services, réseaux et équipements pouvant être gérés par le RGT.

- Réseaux de zone c.-à-d. WAN, MAN et LAN.
- Réseaux à commutation par paquets, par exemple : X.25.
- Réseaux téléphoniques privés, par exemple : PABX.
- Systèmes d'exploitation et leurs périphériques.
- Applications informatiques fonctionnant dans les ordinateurs centraux.
- Ordinateurs centraux.

En outre, un RGT peut être utilisé pour la gestion d'entités et de services répartis que l'on obtient en groupant les éléments énumérés ci-dessus

4. Objectifs.

A partir d'un certain nombre de systèmes de gestion, les opérateurs ont la possibilité de gérer une grande diversité d'équipements, de réseaux et de services à structure répartie.

5. Fonctionnalité.

La fonctionnalité du RGT se compose des éléments suivants :

- aptitude à échanger des informations de gestion avec l'environnement géré;
- aptitude à convertir l'information de gestion de manière à ce que l'information circulant au sein du RT soit cohérente;
- aptitude à transférer de l'information entre sites du RGT;
- aptitude à analyser l'information et à y réagir de façon appropriée;
- aptitude à rendre l'information utile et/ou significative pour l'utilisateur;
- aptitude à remettre l'information de gestion à l'utilisateur de cette information et ce sous une forme appropriée;
- aptitude à garantir un accès sûr à l'information de gestion pour les utilisateurs.

6. Architecture.

L'architecture du RGT doit, entre autres, permettre de gérer des réseaux hybrides, composés d'équipements de réseaux mixtes. Trois aspects fondamentaux en composent l'architecture générale. Ce sont :

- l'architecture fonctionnelle qui décrit la répartition des éléments fonctionnels afin de permettre la création de blocs de fonctions. Ces blocs de fonctions sont à la base de la mise en oeuvre du RGT.

- L'architecture d'information qui se base sur une approche orientée objet et met en correspondance les principes de gestion OSI et les principes du RGT. De plus, elle permet l'adaptation des principes de gestion OSI à l'environnement RGT.
- L'architecture physique qui décrit les interfaces réalisables et donne des exemples de composants physiques du RGT.

7. Architecture fonctionnelle.

L'architecture fonctionnelle du RGT est construite sur un certains nombre de blocs de fonctions RGT. Ceux-ci fournissent les fonctions générales RGT qui permettent à un RGT d'exécuter les fonctions de gestion. L'échange des informations entre ces blocs se fait au moyen de la fonction de communication de données (DCF - Data Communication Function) au travers d'un "point de référence" (sorte de SAP).

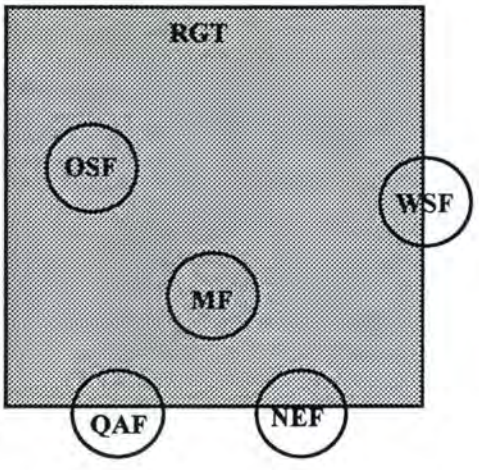


Figure B.1 : Blocs de fonctions du RGT

La figure B.1 illustre le positionnement des différents blocs de fonctions au sein du RGT. Il faut noter la présence de bloc placés à la frontière du RGT car en interaction avec l'extérieur.

Le bloc de fonctions de système d'exploitation (OSF - Operations Systems Function) traite les informations relatives à la gestion des télécommunications pour surveiller/coordonner et/ou commander les fonctions de télécommunications, y compris les fonctions de gestion.

Le bloc de fonctions d'élément de réseau (NEF - Network Element Function) communique avec le RGT afin d'être surveillé et/ou commandé. En outre, il fournit les fonctions de télécommunications et supports nécessaires au réseau de télécommunications sur lequel porte la gestion.

Le bloc de fonctions de poste de travail (WSF - WorkStation function) interprète l'information RGT pour les besoins de l'utilisateur de l'information.

Le bloc de fonctions de médiation (MF - Mediation Function) agit sur les informations passant entre une fonction OSF et une fonction NEF pour faire en sorte que ces informations répondent à l'attente de l'autre partie.

Enfin le bloc de fonctions d'adaptateur Q (QAF - Q Adaptator Function) sert à connecter, dans le cadre du RGT, les entités non-RGT qui sont du type NEF et OSF. (Sorte de traducteur, de proxy).

Comme nous l'avons dit, chacun de ces blocs se compose de plusieurs fonctions qui seront les éléments opérationnels du RGT. Terminons ce point en signalant qu'il existe différentes classes de points de référence en fonction des blocs de fonctions liés; de plus, la fonction DCF utilisée pour l'échange inter-blocs a pour rôle essentiel la mise à disposition des mécanismes de transport de l'information.

Annexe C : Questionnaire type.

1. ARCHITECTURE :
   a- Is your system + distributed? YES -- NO
   + Client/Server? YES -- NO

   b- Under which O.S. runs your system? UNIX (version .....)
   Windows
   Windows NT
   Another One : .....

   d- If your system is distributed, which kind of network (i.e. LAN 802.3) is there between the management stations?

2. DOMAIN :
   a- Which kind of Network are manageable? LAN : .....
   MAN :.....
   WAN :.....
   Voice : .....

   b- Are systems manageable? YES -- NO
   Which ones? system manageable : .....

   c- Are DataBase manageable? YES -- NO
   Which ones? .....

3. EXTENSIBILITY :
   a- How is it possible to be compatible with new features ? .....

   b- How to add new applications (i.e. Performance analysis application written by users)? .....

   c- How to create new applications? (languages) .....

   d- Availability of developpers tool kit? YES -- NO
   Utilisation (what's possible with it?) .....

4. NETWORK (MANAGEMENT) INTERFACE :
   a- Which network management interface are available with your system?
   SNMP
   SNMP v2
   CMIS/P
   another one : .....

   b- Which network interface are available with your system? (i.e. X.25, ATM, Ethernet,.....)
   .....

5. COMPATIBILITY WITH OTHER STANDARDS :
   With which standards is your system compatible? OSF/DME
   OMNIPoint
   Another one : .....

6.DATA MANAGEMENT :
   Which DBMS is useable to manage your management data ? (i.e. SQL,..).....

7. USER INTERFACE:
   a- Is it a graphic user interface? YES -- NO

b- Is it possible to have all the enterprise networks (voice - X.25 - LAN's...) on the same windows?
YES -- NO

c- Is it possible to make a map? YES -- NO

d- Is it possible to make a zoom on the different kind of views? Are the views hierarchical?
YES -- NO

e- Is it possible to have a view of the
Logical TOPOLOGY?
Physical
Organisational (per departement)

f- Which kinds of view are available? .....

g- Is it possible to customize views?
for Each User (his own views)
for the Administrator

h- How is it possible?
Point and click
programming
Another method : .....

i- Which kind of message can the user receive?
Icon change
Textual with coloured background
Sound
another one : .....

j- Is it possible to customize his messages?
for Each User
for the Administrator

k- How is it possible?
Point and click
programming
Another method : .....

8. FUNCTIONNALITIES :
Are these functionnalities available on your system?

a- Autodiscovery YES -- NO

b- Auto Configuration at the installation YES -- NO

c- Update in real time of all the views when there are changes somewhere on the network
YES -- NO

d- Alarm filtering YES -- NO

e- How is it possible to create filters?
Point and Click
Programming
Another Way : .....

f- Is trouble-ticketing (TT) available? YES -- NO

g- How are TT created?
Automatic
Manually
Another Way : .....

h. How is it possible to use TT?
Automatic display on each alarm message
Used for corelation on each alarm message
Used as knowledges base
Another use : .....

i- Is your system doing proactive observation (before alarm - alarm prevention)
j- Is your system doing alarm diagnostic? YES -- NO

h- How is your system doing alarm diagnostic? (i.e. expert system,....) .....

k- Is it possible for the users to modify remotely the configuration of almost devices?
YES -- NO

l- Is it possible for the users to monitor in real time the performances, throughput,.... of the enterprise networks? YES -- NO

m- How is it possible?
gauges and counters
textual display
indications on map
pie chart, histogram,....
another way : .....

n- Is it possible to create performance and utilisation reports?
YES -- NO

o- Which kind of report is possible?
Display
Hard copy
Excel or Lotus compatible
another kind :.....

p- Is security management of the managed networks available?
YES -- NO

q- Which are the main security functionnalities?.....

r- Is user access control available (different level of access which are customizable)?
YES -- NO

s- Is accounting management available? YES -- NO

t- Is your system multivendor compatible (with network OS and/or network management system)?
YES -- NO

u- Are APIs available? YES -- NO

v- Which languages is necessary to create API on your system? .....

9. INSTRUCTION :

How many time do you need to train		
	users	.....Hours or .....days
	administrator	.....Hours or .....days
	developpers	.....Hours or .....days

10. HARDWARE and SOFTWARE (optional answer) :
a - Could you give a description of the different software elements in your products (their functionnalities i.e. network management interface, DBMS modules,...) ?

b- Can you give the standard configuration of the management station and / or server and clients) needed to manage the following networks: "Description of your Managed Networks".

Annexe D: OSF/DME

1. Définition de l'environnement distribué de gestion (Distibuted Management Environnement - DME).

L'environnement distribué de gestion comprend deux composants:

- un ensemble de services d'application qui fournit quelques fonctions de gestion de système parmi les plus critiques;
- une structure comprenant les modules nécessaires au développement d'applications de gestion de divers systèmes.

2. Services d'application.

DME offre des services d'applications distribués sous forme de modules et d'API qui reprennent les tâches de gestion les plus cruciales dans les environnements distribués actuels. Il offre aussi les applications de gestion utilisant ces services ainsi qu'une interface utilisateur. Encore une fois, cette approche modulaire facilite les mises à jour par les différents constructeurs. Ces applications sont :

- gestion de logiciel (mises à jour des logiciels, installation);
- gestion des licences;
- services d'impression.

3. Architecture de DME.

Dans la gestion de systèmes et de réseaux, l'administrateur ne gère qu'en modifiant les informations relatives à des ressources ou des services et en exécutant certaines opérations sur certains services et certaines données. DME regroupe les informations et les opérations au sein d'objets, ce qui offre ainsi une approche structurée. Ceci permet d'exécuter toutes les opérations de gestion suivant le même style d'interaction et la même interface, c. -à- d. en communiquant avec des objets. L'architecture de DME est donc orientée objet et est composée d'un ensemble de modules facilitant le développement de nouvelles applications ou la mise à jour d'anciennes. On retrouve ainsi :

- des services d'objets;
- une interface utilisateur;
- des services de gestion;
- des protocoles de gestion.

3.1. Services d'objets.

Divers éléments font partie de ces services:

- Les deux intermédiaires de requêtes de gestion sont les éléments centraux de DME et ils facilitent la communication entre les applications et les objets. Ils acceptent les requêtes sur les objets, localisent les objets et leur transmettent la requête. Le premier supporte les standards de gestion de réseaux, SNMP et CMIP. Le second permet l'utilisation d'un protocole basé sur RPC.
- Les deux types de serveur d'objets conservent les objets: le premier ne les conserve que pour les opérations à court terme, tandis que le second sert pour les opérations prenant plus de temps.
- Les services de gestion d'évènements: ils permettent la notification d'évènements éventuellement après leur passage au travers de filtres définis au moyen d'un langage de haut niveau.
- Les services de gestion de données: ils ont pour but de permettre la conservation, sur disque, des informations contenues dans les objets.

3.2. Interface utilisateur.

L'interface utilisateur est basée sur OSF/Motif et utilise un gestionnaire d'affichage DME qui interprète les définitions (apparence et possibilité d'action) contenues dans les objets. Il est possible d'avoir diverses vues topologiques personnalisées.

3.3. Services de gestion.

Les services de gestion permettent les définitions de politiques et de modèles de gestion flexibles et personnalisables.

3.4. Protocoles de gestion.

DME supporte l'utilisation des protocoles SNMP, CMIP et DCE-RPC pour la communication entre objets et applications.

4. Boîte à outils de développement.

Pour permettre le développement d'applications de gestion, une boîte à outils composée d'API avec l'architecture DME est disponible.

BIBLIOGRAPHIE.

[ABEC93] Abeck, S., Clemm, A. & Holberg, U.,
"Simply Open Network Management :
An approach for the integration of SNMP into OSI Management Concepts",
Proceedings of the IFIP TC6/WG 6.6 symposium on Integrated Network Management,
Elsevier Science Publishers B.V. (North Holland),
Amsterdam, 1993,
ISBN 0 444 89982 0
pp 361-374.

[AETC93] HQ AETC/SCTT (R., PFAFFINGER),
"Network Management System evaluation, HP OpenView 3.1",
Document interne, non-publié,
USA, 1993.

[ALCA94] ALCATEL BELL,
"Network and Services management seminar : Introduction"
Documents de séminaire, non publié,
ALCATEL BELL,
Bruxelles, 1994.

[BHUS94] Bhushan, B., Patel,A., De Souza,J.N. & Claudé J.P.,
"Managing Heterogeneous Networks - Integrator Based Approach -"
Proceedings of the IFIP TC6/WG 6.4 symposium on
Advanced Information Processing techniques for LAN and MANAGEMENT Management,
Elsevier Science Publishers B.V. (North Holland),
Amsterdam, 1994,
ISBN 0 444 81634 8
pp 129-144.

[BODA95] Bodart, F.,
"Interface homme-machine",
Notes de cours, Non-publié,
FUNDP,
Namur, 1995.

[BULL94] BULL
"Integrated System Management v3",
Document interne, Non-publié,
1994

[CASE90] Case, J., Fedor, M., Schoffstall, M. & Davin, J.,
"A Simple Network Management Protocol (SNMP)"
RFC 1157,
Network Working Group,
USA, 1990.

[CHIP94] Chipman, T. & Chandler, G.,
"Managing the Branch Office : Part1"
Network Support Encyclopedia, Novell Research, (CD-ROM)
Novell Inc.,
USA, 1994.

[DATS93] DATSA Belgium sprl,
"Network Management techniques - An overview"
Telinfo High Tech Institute,
Belgique, 1993.

[DAVI92] Davin, J., Galvin, J., & McCloghrie, K.,
"SNMP Administrative Model",
RFC 1351
Network Working Group,
USA, 1992

[DISA93] Disabato, C.M.,
"Key Technologies for Integrated Network Management",
Elsevier Science Publishers B.V. (North Holland),
Amsterdam, 1993,
ISBN 0 444 89982 0
pp 423-434.

[FCEM94] FCEMT (Future Computing Environment Monitoring Team),
"Final report",
Non-publié,
University of Michigan, 1994

[GALV93] Galvin, J. & McCloghrie, K.
"Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)"
RFC 1445
Network Working Group
USA, 1993

[JAVA94-1] Javaux, D.,
"Psychologie cognitive: Ergonomie cognitive de l'interaction homme-machine",
Notes de cours, Non-publié,
FUNDP,
Namur, 1994.

[JAVA94-2] Javaux, D.,
"Psychologie cognitive: Supervisory control",
Notes de cours, Non-publié,
FUNDP,
Namur, 1994.

[LARO59] Petit Larousse,
Larousse,
France, 1959,

[LESU95] Lesuisse, R.,
"Théorie des organisations : gestion de projets informatiques -
Chapitre 3 : la gestion du risque",
Notes de cours, Non-publié,
FUNDP,
Namur, 1995.

[LOBE94] Lobet - Maris, C.,
"Théorie des organisations : Approche politique des structures d'organisation",
Notes de cours, Non-publié,
FUNDP,
Namur, 1994.

[MAHA94] Mahamat, G., Das, A., & Bochmann, G.v.,
"An Overview of Fault Management in Telecommunication Networks",
Proceedings of the IFIP TC6/WG 6.4 symposium on
Advanced Information Processing techniques for LAN and MANAGEMENT Management,
Elsevier Science Publishers B.V. (North Holland),
Amsterdam, 1994,
ISBN 0 444 81634 8
pp 69-85.

[MAZU93] Mazumdar, S., Brady, S. & Levine, D.L.,
"Design of Protocol Independent Mnagement Agent to support SNMP and CMIP queries",
Elsevier Science Publishers B.V. (North Holland),
Amsterdam, 1993,
ISBN 0 444 89982 0
pp 377-388.

[MCCL91] McCloghrie, K & Rose, M..,
"Management Information Base for Network Management of TCP/IP-based internets : MIB-II"
RFC 1213,
Network Working Group,
USA, 1990.

[MEIN91] Meinadier, J-P.,
"L'interface utilisateur - Pour une informatique plus conviviale",
DUNOD,
Paris, 1991
ISBN 2-10-000160-4

[MERL95] Merlant, P.,
"Editorial : Intégration",
L'expansion Management Review,
Mars, 1995,
p 3.

[MURR93] Murril, B.,
"OMNIPoint: An Implementation Guide to Integrated Networked Information Systems Management",
Elsevier Science Publishers B.V. (North Holland),
Amsterdam, 1993,
ISBN 0 444 89982 0
pp 405-417.

[NACH95] Nachtergaele, V.,
"Cours de télécommunications. Chapitre relatif à SNMP.
Description d'un protocole de gestion de réseau : le protocole SNMP",
Non publié,
FUNDP,
Namur, 1995.

[NOVE93] Novell, Inc.,
"NetWare Buyer's Guide",
Novell, Inc.,
Volume V, N°2,
USA, 1993.

[PENN94] Penna, C. & De Souza, J.N.,
"Unification of Heterogeneous Network Management",
Proceedings of the IFIP TC6/WG 6.4 symposium on
Advanced Information Processing techniques for LAN and MANAGEMENT Management,
Elsevier Science Publishers B.V. (North Holland),
Amsterdam, 1994,
ISBN 0 444 81634 8
pp 117-128.

[ROSE90] Rose, M. & McCloghrie, K.,
"Structure & Identification of Management Information for TCP/IP based Internets"
RFC 1155,
Network Working Group,
USA, 1990.

[STAL94] Stallings, W.,
" SNMP, SNMPv2, and CMIP : the practical guide to network management standards ",
Addison-Wesley,
Readings, 1994,
ISBN 0-201-63331-0.

[TERP92] Terplan, K,
"Communication Network Management",
2°edition,
Prentice Hall,
Englewood Cliffs - New Jersey, 1992,
ISBN 0-13-156449-8.

[WALD91] Waldbusser, S.,
" Remote Network Monitoring Management Information Base",
RFC 1271,
Network Working Group
USA, 1991.

[WIER93] Wiersinga, H.A.,
"Korte metten met eilandautomatisering",
Telecommagazine,
Novembre, 1993,
pp 28-34.

[ZNAT94] Znaty, S. & Sclavos, J.,
"Annotated Bibliography on Network Management",
Computer Communication Review,
Vol 24, N°1,
USA, 1994,
pp 37-56.

Sites Internet intéressants.

Network Management Server (NMS)	= "http://smurfland.cit.buffalo.edu/NetMan/index.html"
Experimental IETF RFCs Index	= "http://lurch.cit.buffalo.edu/Misc/RFC/"
International Telecommunication Union	= "http://www.itu.ch:80/"
AETC Base Network Control Center Page	= "http://www.aetc.af.mil/AETC-NetMgmt/NMS-mainmenu.html"
Network Management Forum Home Page	= "http://www.nmf.org/"
Strategic Information Resources	= "http://iquest.com/~nmuller/"
DME Overview	= "http://www.osf.org/comm/lit/OSF-DME-PD-0394-2.html"
Network Management Resources and Information	="http://www.wp.com/lowens/"
Network and Systems management sites/references: European (Micromuse)	="http://www.micromuse.co.uk/netman/"
The SimpleWeb	="http://snmp.cs.utwente.nl "